

Package: MCMCvis (via r-universe)

September 18, 2024

Type Package

Title Tools to Visualize, Manipulate, and Summarize MCMC Output

Version 0.16.4

Maintainer Casey Youngflesh <caseyyoungflesh@gmail.com>

Description Performs key functions for MCMC analysis using minimal code - visualizes, manipulates, and summarizes MCMC output. Functions support simple and straightforward subsetting of model parameters within the calls, and produce presentable and 'publication-ready' output. MCMC output may be derived from Bayesian model output fit with 'Stan', 'NIMBLE', 'JAGS', and other software.

License GPL-3

URL <https://github.com/caseyyoungflesh/MCMCvis>

BugReports <https://github.com/caseyyoungflesh/MCMCvis/issues>

LazyData TRUE

Imports coda, rstan, methods, overlapping, colorspace

Depends R (>= 3.5.0)

RoxygenNote 7.3.1

Suggests knitr, rmarkdown, testthat, cmdstanr (>= 0.5.0), posterior

Additional_repositories <https://mc-stan.org/r-packages/>

VignetteBuilder knitr

Encoding UTF-8

Repository <https://caseyyoungflesh.r-universe.dev>

RemoteUrl <https://github.com/caseyyoungflesh/mcmcvis>

RemoteRef HEAD

RemoteSha 97a5e41edb515ac48f0630b340d3b5e18c96c7e8

Contents

MCMCvis-package	2
MCMCchains	3
MCMCdiag	5
MCMCplot	7
MCMCpstr	12
MCMCsummary	13
MCMCtrace	16
MCMC_data	20
MCMC_data2	21
Index	22

MCMCvis-package	<i>The 'MCMCvis' package</i>
-----------------	------------------------------

Description

'MCMCvis' is an R package used to visualize, manipulate, and summarize MCMC output. MCMC output may be derived from Bayesian model output fit with JAGS, Stan, or other MCMC samplers.

Details

The following functions are currently available:

- MCMCsummary - summarize MCMC output for particular parameters of interest
- MCMCpstr - summarize MCMC output for particular parameters of interest while preserving original parameter structure
- MCMCtrace - create trace and density plots of MCMC chains for particular parameters of interest
- MCMCchains - easily extract posterior chains from MCMC output for particular parameters of interest
- MCMCplot - create caterpillar plots from MCMC output for particular parameters of interest)
- MCMCdiag - create a .txt file and save specified objects that summarize model inputs, outputs, and diagnostics

Example data can be loaded using `data(MCMC_data)`.

'MCMCvis' was designed to perform key functions for MCMC analysis using minimal code, in order to free up time/brainpower for interpretation of analysis results. Functions support simple and straightforward subsetting of model parameters within the calls, and produce presentable, 'publication-ready' output.

The vignette can be run using `vignette('MCMCvis')` if vignette is built when installing package.

Author(s)

Casey Youngflesh <caseyyoungflesh@gmail.com> Christian Che-Castaldo <chriscecastaldo@protonmail.com>

Author(s)

Maintainer: Casey Youngflesh <caseyyoungflesh@gmail.com> ([ORCID](#))

Authors:

- Christian Che-Castaldo <christian.che-castaldo@stonybrook.edu> ([ORCID](#))

Other contributors:

- Tyler Hardy [contributor]

See Also

Useful links:

- <https://github.com/caseyyoungflesh/MCMCvis>
- Report bugs at <https://github.com/caseyyoungflesh/MCMCvis/issues>

MCMCchains

Extract posterior chains from MCMC output

Description

Extract posterior chains from MCMC output for specific parameters of interest.

Usage

```
MCMCchains(  
  object,  
  params = "all",  
  excl = NULL,  
  ISB = TRUE,  
  exact = TRUE,  
  mcmc.list = FALSE,  
  chain_num = NULL  
)
```

Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters of interest. Default 'all' returns chains for all parameters.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored. If FALSE, square brackets are not ignored. This allows partial names to be used when specifying parameters of interest. Use exact argument to specify whether input from params and excl arguments should be matched exactly.

<code>exact</code>	Logical specifying whether input from <code>params</code> and <code>excl</code> arguments should be matched exactly (after ignoring square brackets if <code>ISB = FALSE</code>). If <code>TRUE</code> , input from <code>params</code> and <code>excl</code> are matched exactly (after taking <code>ISB</code> argument into account). If <code>FALSE</code> , input from <code>params</code> and <code>excl</code> are matched using regular expression format (after taking <code>ISB</code> argument into account).
<code>mcmc.list</code>	Logical specifying whether to return an <code>mcmc.list</code> . If <code>TRUE</code> , an <code>mcmc.list</code> object is returned, rather than a matrix.
<code>chain_num</code>	Numeric - specifies posterior chain number. When a value is specified, posterior for only that chain is output. Useful for determining the last iteration for each parameter, to be used as initial values in a subsequent model, to effectively 'continue' a model run.

Details

Function returns matrix with one parameter per column (for specified parameters). Each iteration is represented as a row. Multiple chains for each parameter are combined to one posterior chain (unless `chain_num` is specified, in which case only the specified chain will be returned). Parameters are arranged in columns alphabetically.

object argument can be a `stanfit` object (`rstan` package), a `CmdStanMCMC` object (`cmdstanr` package), a `stanreg` object (`rstanarm` package), a `brmsfit` object (`brms` package), an `mcmc.list` object (`coda` and `rjags` packages), `mcmc` object (`coda` and `nimble` packages), `list` object (`nimble` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Examples

```
#Load data
data(MCMC_data)

#Extract MCMC chains
ex <- MCMCchains(MCMC_data)
apply(ex, 2, mean)

#Extract MCMC chains for just 'beta' parameters
ex2 <- MCMCchains(MCMC_data, params = 'beta')
apply(ex2, 2, mean)

#Just 'beta[1]', 'beta[4]', and 'alpha[3]'
ex3 <- MCMCchains(MCMC_data, params = c('beta[1]', 'beta[4]', 'alpha[3]'),
                  ISB = FALSE, exact = TRUE)
apply(ex3, 2, sd)
```

Description

Model diagnostics and summary. Function reads information embedded in model fit object. Output varies by model fit object type but includes model run inputs, diagnostic information, and parameter summary. See DETAILS below for more information.

Usage

```
MCMCdiag(
  object,
  file_name,
  dir = getwd(),
  mkdir,
  add_field,
  add_field_names,
  save_object = FALSE,
  params = "all",
  excl = NULL,
  ISB = TRUE,
  exact = TRUE,
  obj_name,
  add_obj,
  add_obj_names,
  cp_file,
  cp_file_names,
  open_txt = TRUE,
  summary = TRUE,
  ...
)
```

Arguments

object	Object containing MCMC output. See DETAILS below.
file_name	Character string with name of .txt file to be saved to dir (or mkdir if specified). If not specified, 'MCMCdiag.txt' will be used.
dir	Character string with directory where file(s) (or directory is argument for mkdir is specified) will be created. Defaults to working directory. An absolute or relative (to the working directory) path can be used.
mkdir	Character string with name of directory to be created. If specified, a directory will be created within the directory specified by dir. If directory exists, '_1' will be appended to specified name.
add_field	Object (or vector of objects) to be added to the .txt file.

<code>add_field_names</code>	Character string (or vector of character strings) specifying the name(s) of the <code>add_field</code> object(s).
<code>save_object</code>	Logical specifying whether the model output provided to the function (object) should be saved as a <code>.rds</code> file to <code>dir</code> (or <code>mkdir</code> if specified). Note that <code>.rds</code> files can be opened with <code>rdsRDS()</code> .
<code>params</code>	Character string (or vector of character strings) denoting parameters of interest for summary. Object to be saved out will be a subset of parameters and will be an <code>mcmc.list</code> object, losing other object types.
<code>excl</code>	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with <code>params</code> argument to select parameters of interest.
<code>ISB</code>	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the <code>params</code> and <code>excl</code> arguments. If <code>TRUE</code> , square brackets are ignored. If <code>FALSE</code> , square brackets are not ignored. This allows partial names to be used when specifying parameters of interest. Use <code>exact</code> argument to specify whether input from <code>params</code> and <code>excl</code> arguments should be matched exactly.
<code>exact</code>	Logical specifying whether input from <code>params</code> and <code>excl</code> arguments should be matched exactly (after ignoring square brackets if <code>ISB = FALSE</code>). If <code>TRUE</code> , input from <code>params</code> and <code>excl</code> are matched exactly (after taking <code>ISB</code> argument into account). If <code>FALSE</code> , input from <code>params</code> and <code>excl</code> are matched using regular expression format (after taking <code>ISB</code> argument into account).
<code>obj_name</code>	Character string specifying the file name of the <code>.rds</code> file (created from object) if <code>save_object = TRUE</code> .
<code>add_obj</code>	List with additional object(s) to be saved as <code>.rds</code> files to <code>dir</code> (or <code>mkdir</code> if specified). Objects can be of any types. Multiple objects can be specified. Note that <code>.rds</code> files can be opened with <code>rdsRDS()</code> .
<code>add_obj_names</code>	Character string (or vector of character strings) specifying the name(s) of the objects to be saved as <code>.rds</code> files, specified with <code>add_obj</code> .
<code>cp_file</code>	Character string (or vector of character strings) specifying file(s) to be copied to <code>dir</code> (or <code>mkdir</code> if specified). Absolute or relative (to the working directory) paths can be used.
<code>cp_file_names</code>	Character string (or vector of character strings) specifying new names for files to be copied specified by <code>cp_file</code> . If no argument is provided, the copy names will be identical to the originals.
<code>open_txt</code>	Logical - if <code>open_txt = TRUE</code> <code>.txt</code> file will open in default <code>.txt</code> viewer after being generated.
<code>summary</code>	Logical specifying whether or not to output summary information from MCMCsummary at the bottom of the <code>.txt</code> file.
<code>...</code>	Arguments to be passed to MCMCsummary when generating summary if <code>summary = TRUE</code> .

Details

Some diagnostic information is only provided for models fit with particular pieces of software. For example, `rstan` output includes additional diagnostics related to the NUTS sampler. Output from

jagsUI includes runtime information, but output from rjags does not. Note that this information could be fed manually to the function using the `add_field` argument.

object argument can be a `stanfit` object (rstan package), a `CmdStanMCMC` object (cmdstanr package), a `stanreg` object (rstanarm package), a `brmsfit` object (brms package), an `mcmc.list` object (coda and rjags packages), `mcmc` object (coda and nimble packages), `list` object (nimble package), an `R2jags` model object (R2jags package), a `jagsUI` model object (jagsUI package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Output presented in `.txt` file varies by model fit object type but includes: model run time, number of warmup/burn-in iterations, total iterations, thinning rate, number of chains, specified adapt delta, specified max tree depth, specific initial step size, mean accept stat, mean tree depth, mean step size, number of divergent transitions, number max tree depth exceeds, number of chains with BFMI warnings, max Rhat (maximum Rhat of any parameter printed), min n.eff (minimum n.eff of any parameter printed), parameter summary information (passed from `MCMCsummary`), and any additional information fed to the `add_field` argument. See documentation for specific software used to fit model for more information on particular diagnostics.

Examples

```
#Load data
data(MCMC_data)

# MCMCdiag(MCMC_data,
#          #name of .txt file to be saved
#          file_name = 'blog-model-summary-2021-01-15',
#          #directory within which to save .txt file
#          dir = '~/Desktop',
#          #round MCMCsummary output in .txt file to two digits
#          round = 2,
#          #add two fields to .txt file
#          add_field = c(50, '1.0'),
#          #names of two additional fields to add to .txt file
#          add_field_names = c('Time (min)', 'Data version'))
```

Description

Visualize posterior distributions from MCMC output for specific parameters of interest using caterpillar plots. Color of median dot represents the overlap of the posterior distribution with 0 (or other specified value).

Usage

```

MCMCplot(
  object,
  object2 = NULL,
  params = "all",
  HPD = FALSE,
  ci = c(50, 95),
  excl = NULL,
  ISB = TRUE,
  exact = TRUE,
  ref = 0,
  ref_ovl = FALSE,
  col = "black",
  col2 = "red",
  offset = 0.1,
  rank = FALSE,
  horiz = TRUE,
  xlim,
  ylim,
  xlab,
  ylab,
  main,
  labels,
  guide_lines = FALSE,
  guide_axis = TRUE,
  sz_labels = 1.2,
  sz_med = 1.5,
  sz_thick = 5,
  sz_thin = 2,
  sz_ax = 3,
  sz_ax_txt = 1.3,
  sz_tick_txt = 1.2,
  sz_main_txt = 1.2,
  pos_tick,
  mar = c(5.1, 4.1, 4.1, 2.1)
)

```

Arguments

object	Object containing MCMC output. See DETAILS below.
object2	Optional second object containing MCMC output. If specified, parameter estimates from each model will be displayed in a paired manner. Parameter names for 'object' and 'object2' must be identical. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters to be plotted. Default 'all' plots posteriors for all parameters. See VALUE below.
HPD	Logical specifying whether to calculate equal-tailed credible intervals (HPD =

	FALSE) or highest posterior density intervals (HPD = TRUE) for the selected parameters. Default is HPD = FALSE.
ci	Numeric vector of length 2, where each element is (0,100] and represents the width of an equal-tailed (HPD = FALSE) or highest posterior density (HPD = TRUE) credible interval. The first element of this vector corresponds to the thicker (narrower) credible interval displayed on the plot (default is 0.5) and the second element of this vector corresponds to the thinner (wider) credible interval (default is 0.95). The first credible interval width (ci[1]) must be less than or equal to the width of the second credible interval (ci[2]).
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored. If FALSE, square brackets are not ignored. This allows partial names to be used when specifying parameters of interest. Use exact argument to specify whether input from params and excl arguments should be matched exactly.
exact	Logical specifying whether input from params and excl arguments should be matched exactly (after ignoring square brackets if ISB = FALSE). If TRUE, input from params and excl are matched exactly (after taking ISB argument into account). If FALSE, input from params and excl are matched using regular expression format (after taking ISB argument into account).
ref	Value indicating where vertical reference line should be created and what value to use a reference for caterpillar median coloration. Default is ref = 0. Argument NULL will plot no reference line.
ref_ovl	Logical specifying whether the style/color of plotted median dots and CI should be changed based on whether the specified credible intervals (50 % and 95 % by default) overlap the reference line. See DETAILS for more information.
col	Character string (or vector of character strings) specifying which color to render estimates on plot. When ref_ovl = TRUE, this argument has no effect and colors plotted will be based on the credible intervals and reference line. Number of specified colors must equal the number of specified parameters or one.
col2	Character string (or vector of character strings) specifying which color to render estimates on plot for object2 (if specified). Number of specified colors must equal the number of specified parameters or one. Red by default.
offset	Value indicating how much to offset plotted posteriors when object2 is specified (i.e., control the amount of space between the two plotted posteriors for each parameter). The distance from one set of parameters to another corresponds to a value of 1.
rank	Logical specifying whether output should be ranked. If TRUE posteriors will be ranked in decreasing order (based on specified measure of centrality) from top down.
horiz	Logical specifying orientation of plot. If TRUE posteriors will be plotted running horizontally (parallel to the x-axis). If FALSE posteriors will be plotted running vertically (perpendicular to the x-axis).

<code>xlim</code>	Numerical vector of length 2, indicating range of x-axis. Only applicable if <code>horiz = TRUE</code> .
<code>ylim</code>	Numerical vector of length 2, indicating range of y-axis. Only applicable if <code>horiz = FALSE</code> .
<code>xlab</code>	Character string labeling x-axis. Only applicable if <code>horiz = TRUE</code> . Default label is 'Parameter Estimate'. Option <code>NULL</code> will return plot with no label on x-axis.
<code>ylab</code>	Character string labeling y-axis. Only applicable if <code>horiz = FALSE</code> . Default label is 'Parameter Estimate'. Option <code>NULL</code> will return plot with no label on y-axis.
<code>main</code>	Character string indicating title of plot.
<code>labels</code>	Character string (or vector of character strings if plotting > 1 parameter) labeling parameter estimates along y-axis (if <code>horiz = FALSE</code>) or x-axis (if <code>horiz = TRUE</code>). Default option will use parameter names from object. Option <code>NULL</code> will return plot with no labels on axis.
<code>guide_lines</code>	Logical specifying whether to plot reference lines for each parameter in order to better visualize which parameter names correspond to each posterior.
<code>guide_axis</code>	Logical specifying whether a second axis should be plotted (x-axis if <code>HORIZ = TRUE</code> , y-axis if <code>HORIZ = FALSE</code>) to help interpret values on plot.
<code>sz_labels</code>	Number specifying size of text for parameter labels on axis.
<code>sz_med</code>	Number specifying size of points represents posterior medians.
<code>sz_thick</code>	Number specifying thickness of 50 percent CI line (thicker line).
<code>sz_thin</code>	Number specifying thickness of 95 percent CI line (thinner line).
<code>sz_ax</code>	Number specifying thickness of axis and ticks.
<code>sz_ax_txt</code>	Number specifying size of text for axis label.
<code>sz_tick_txt</code>	Number specifying size of text for tick labels on axis.
<code>sz_main_txt</code>	Number specifying size of text for main title.
<code>pos_tick</code>	Numeric vector specifying where ticks on axis should be placed.
<code>mar</code>	Numerical vector of length 4 specifying plot margins - (<code>BOTTOM</code> , <code>LEFT</code> , <code>TOP</code> , <code>RIGHT</code>). Changes to the margin should be made within the function rather than using the <code>par</code> call. Default is <code>c(5.1, 4.1, 4.1, 2.1)</code> - the R plot default.

Details

Points represent posterior medians. Parameters where the smaller specified credible intervals (first value in vector provided to `ci` argument; 50% by default) overlap 0 (or other value specified by the `ref` argument) are indicated by 'open' circles and a lighter color than that specified (e.g., gray will be displayed with the default black is specified for `col`). Parameters where the specific smaller credible intervals DO NOT overlap 0 (or other specified value) AND the larger specified credible intervals (second value in vector provided to `ci` argument; 95% by default) DO overlap 0 (or other specified value) are indicated by 'closed' circles and a lighter color that that specified. Parameters

where the larger specified credible intervals DO NOT overlap 0 (or other specified value) are indicated by 'closed' circles and the color specified (black by default). Thick lines represent the smaller specified credible intervals percent credible intervals (50% by default) while thin lines represent the larger specified credible intervals (95 % by default). `ref_ovl = TRUE` can be used to enable this feature. When two model objects are supplied to the function (with `object` and `object2`) and no argument is supplied to `col` or `col2`, light red lines can be interpreted as analogous to gray lines. When `ref_ovl = TRUE` and a color (or colors) other than the default are specified (with the `col` and `col2` arguments) lighter versions of the color specified are used in black of the light gray and/or light red lines.

When `object2` is specified, paired caterpillar plots of each parameter are produced. For this reason, parameter names of `object` and `object2` specified with the `params` argument must be identical (to be used for comparing posterior estimates of similar models). `col` and `col2` arguments can be specified to change the color of output from `object` and `object2`, respectively. By default, output from `object` is plotted in black and `object2` is plotted in red. The `ref_ovl` argument can also be specified.

`object` argument can be a `stanfit` object (`rstan` package), a `CmdStanMCMC` object (`cmdstanr` package), a `stanreg` object (`rstanarm` package), a `brmsfit` object (`brms` package), an `mcmc.list` object (`coda` and `rjags` packages), `mcmc` object (`coda` and `nimble` packages), `list` object (`nimble` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Notes

When specifying `rank = TRUE` and specifying labels for `labels`, labels will be applied to parameters before they are ranked.

Thanks to Cinner et al. 2016, whose Fig. 1 inspired this plot.

References

Cinner, J. E., C. Huchery, M. A. MacNeil, N. A. J. Graham, T. R. McClanahan, J. Maina, E. Maire, J. N. Kittinger, C. C. Hicks, C. Mora, E. H. Allison, S. D'Agata, A. Hoey, D. A. Feary, L. Crowder, I. D. Williams, M. Kulbicki, L. Vigliola, L. Wantiez, G. Edgar, R. D. Stuart-Smith, S. A. Sandin, A. L. Green, M. J. Hardt, M. Beger, A. Friedlander, S. J. Campbell, K. E. Holmes, S. K. Wilson, E. Brokovich, A. J. Brooks, J. J. Cruz-Motta, D. J. Booth, P. Chabanet, C. Gough, M. Tupper, S. C. A. Ferse, U. R. Sumaila, and D. Mouillot. 2016. Bright spots among the world's coral reefs. *Nature* 535:416-419.

Examples

```
#Load data
data(MCMC_data)

#Plot MCMC output
MCMCplot(MCMC_data, labels = NULL)

#Just 'beta' parameters
MCMCplot(MCMC_data, params = 'beta')
```

```

#Just 'beta' parameters using highest posterior density intervals
MCMCplot(MCMC_data, params = 'beta', HPD = TRUE)
#Just 'beta[1]', 'beta[4]', and 'alpha[3]'
MCMCplot(MCMC_data, params = c('beta[1]', 'beta[4]', 'alpha[3]'), ISB = FALSE, exact = TRUE)

#Just 'beta[1]', 'beta[4]', and 'alpha[3]' and change the credible interval widths
MCMCplot(MCMC_data, ci = c(50, 89), params = c('beta[1]', 'beta[4]', 'alpha[3]'),
  ISB = FALSE, exact = TRUE)

#Rank parameters by posterior mean
MCMCplot(MCMC_data, params = 'beta', rank = TRUE)

#Create vertical plot
MCMCplot(MCMC_data, params = 'beta', horiz = FALSE)

```

MCMCpstr

Summarize and extract posterior chains from MCMC output while preserving parameter structure

Description

Extract summary information and posterior chains from MCMC output (specific function specified) for specific parameters of interest while preserving original parameter structure (i.e., scalar, vector, matrix, array). Function outputs a list with calculated values or posterior chains for each specified parameter.

Usage

```

MCMCpstr(
  object,
  params = "all",
  excl = NULL,
  ISB = TRUE,
  exact = TRUE,
  func = mean,
  type = "summary"
)

```

Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters to be returned in output. Default 'all' returns all parameters in output.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.

ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the <code>params</code> and <code>excl</code> arguments. If TRUE, square brackets are ignored. If FALSE, square brackets are not ignored. This allows partial names to be used when specifying parameters of interest. Use <code>exact</code> argument to specify whether input from <code>params</code> and <code>excl</code> arguments should be matched exactly.
<code>exact</code>	Logical specifying whether input from <code>params</code> and <code>excl</code> arguments should be matched exactly (after ignoring square brackets if <code>ISB = FALSE</code>). If TRUE, input from <code>params</code> and <code>excl</code> are matched exactly (after taking <code>ISB</code> argument into account). If FALSE, input from <code>params</code> and <code>excl</code> are matched using regular expression format (after taking <code>ISB</code> argument into account).
<code>func</code>	Function to be performed on MCMC output. When output of specified function is greater than length 1, an extra dimension is added. For instance, output of length 3 for a parameter with dimensions 2x2 results in a 2x2x3 output. Functions that produce output with dimensionality greater than 1 are not permitted. <code>func</code> is ignored when <code>type = 'chains'</code> .
<code>type</code>	Character string specifying whether to return summary information (calculated based on <code>func</code> argument) or posterior chains. Valid options are <code>'summary'</code> and <code>'chains'</code> . When <code>type = 'chains'</code> , the <code>'func'</code> argument is ignored. When <code>type = 'chains'</code> , posterior chains are concatenated and stored in the last dimension in the array for each element (parameter) of the list.

Details

object argument can be a `stanfit` object (`rstan` package), a `CmdStanMCMC` object (`cmdstanr` package), a `stanreg` object (`rstanarm` package), a `brmsfit` object (`brms` package), an `mcmc.list` object (`coda` and `rjags` packages), `mcmc` object (`coda` and `nimble` packages), `list` object (`nimble` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Examples

```
#Load data
data(MCMC_data)

MCMCpstr(MCMC_data, func = function(x) quantile(x, probs = c(0.01, 0.99)))
```

MCMCsummary

Summarize MCMC output

Description

Extract summary information from MCMC output (mean, median, quantiles, Gelman-Rubin convergence statistic, number of effective samples, and specified custom metrics) for specific parameters of interest.

Usage

```
MCMCsummary(
  object,
  params = "all",
  excl = NULL,
  ISB = TRUE,
  exact = TRUE,
  probs = c(0.025, 0.5, 0.975),
  hpd_prob = 0.95,
  HPD = FALSE,
  pg0 = FALSE,
  digits = NULL,
  round = NULL,
  Rhat = TRUE,
  n.eff = TRUE,
  func = NULL,
  func_name = NULL
)
```

Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters to be returned in summary output. Default 'all' returns all parameters in summary output.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored. If FALSE, square brackets are not ignored. This allows partial names to be used when specifying parameters of interest. Use exact argument to specify whether input from params and excl arguments should be matched exactly.
exact	Logical specifying whether input from params and excl arguments should be matched exactly (after ignoring square brackets if ISB = FALSE). If TRUE, input from params and excl are matched exactly (after taking ISB argument into account). If FALSE, input from params and excl are matched using regular expression format (after taking ISB argument into account).
probs	Numeric vector where each element in (0,1) representing probabilities used to calculate posterior sample quantiles for the selected parameters. Default is c(0.025, 0.5, 0.975).
hpd_prob	Scalar in (0,1) representing probability used to calculate highest posterior density intervals for the selected parameters. Default is 0.95.
HPD	Logical specifying whether to calculate equal-tailed credible intervals (HPD = FALSE) or highest posterior density intervals (HPD = TRUE) for the selected parameters. Default is HPD = FALSE.

<code>pg0</code>	Logical specifying whether to calculate the proportion of the posterior that is greater than 0, rounded to 2 digits.
<code>digits</code>	Number of significant digits to include for posterior summary. All computed digits will be included by default. Note that Rhat is always rounded to 2 decimal places.
<code>round</code>	Number of decimal places to round to for posterior summary. Cannot be used in conjunction with <code>digits</code> argument. Note that Rhat is always rounded to 2 decimal places.
<code>Rhat</code>	Logical specifying whether to calculate and display the potential scale reduction statistic (Rhat). Values near 1 suggest convergence (Brooks and Gelman 1998). <code>Rhat = FALSE</code> will prevent display of this column in summary output. Specifying <code>Rhat = FALSE</code> , may increase function speed for very large <code>mcmc.list</code> objects.
<code>n.eff</code>	Logical specifying whether to calculate and display the number of effective samples for each parameter. <code>n.eff = FALSE</code> will prevent display of this column in summary output. Specifying <code>n.eff = FALSE</code> , may increase function speed for very large <code>mcmc.list</code> objects. Default is <code>n.eff = TRUE</code> .
<code>func</code>	Function to be performed on MCMC output. If a function is specified, it will be evaluated on posteriors for each specified parameter and returned as a column in the summary output (or multiple columns if the function returns more than one value).
<code>func_name</code>	Character string (or vector of character strings) specifying labels for output from <code>func</code> argument. If <code>func_name</code> is not specified, columns with <code>func</code> argument will be labeled 'func'.

Value

Function returns summary information (including parameter posterior mean, posterior sd, quantiles, potential scale reduction statistic (Rhat), number of effective samples, and other specified metrics) for specified parameters.

Details

`object` argument can be a `stanfit` object (`rstan` package), a `CmdStanMCMC` object (`cmdstanr` package), a `stanreg` object (`rstanarm` package), a `brmsfit` object (`brms` package), an `mcmc.list` object (`coda` and `rjags` packages), `mcmc` object (`coda` and `nimble` packages), `list` object (`nimble` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Notes

For `mcmc.list`, `mcmc`, and `list` objects, the potential scale reduction statistic (Rhat) is calculated using the `gelman.diag` function in the `coda` package (what is typically displayed in the summary output from models fit with JAGS). For `stanfit` (as well as `CmdStanMCMC`, `stanreg`, and `brmsfit` objects) and `jagsUI` objects, Rhat is calculated using a 'split chain' Rhat (in their

respective packages), which is thought to be a more conservative diagnostic (Stan Development Team 2018).

For `mcmc.list`, `mcmc`, and `list` objects, the number of effective samples is calculated using the `effectiveSize` function in the `coda` package. For `stanfit` (as well as `CmdStanMCMC`, `stanreg`, and `brmsfit` objects) and `jagsUI` objects, `n.eff` is calculated using a slightly different method of computation for the number of effective samples (Stan Development Team 2018). For `CmdStanMCMC` objects, both bulk and tail `n.eff` is calculated.

References

Brooks, S. P., and A. Gelman. 1998. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics* 7:434.

Stan Development Team. 2018. Stan Modeling Language Users Guide and Reference Manual, Version 2.18.0. <https://mc-stan.org>

Examples

```
#Load data
data(MCMC_data)

#Summary information for MCMC output - display 2 significant digits
MCMCsummary(MCMC_data, digits = 2)

#Just 'beta' parameters - round to 2 decimal places
MCMCsummary(MCMC_data, params = 'beta', round = 2)

#Just 'beta[1]', 'beta[4]', and 'alpha[3]'
MCMCsummary(MCMC_data, params = c('beta[1]', 'beta[4]', 'alpha[3]'), ISB = FALSE, exact = TRUE)
```

MCMCtrace

Trace and density plots from MCMC output

Description

Trace and density plots of MCMC chains for specific parameters of interest. Print plots to pdf by default.

Usage

```
MCMCtrace(
  object,
  params = "all",
  excl = NULL,
  ISB = TRUE,
  exact = TRUE,
  iter = 5000,
  gvals = NULL,
```



```

priors = NULL,
post_zm = TRUE,
PPO_out = FALSE,
Rhat = FALSE,
n.eff = FALSE,
ind = FALSE,
pdf = TRUE,
plot = TRUE,
open_pdf = TRUE,
filename,
wd = getwd(),
type = "both",
ylim = NULL,
xlim = NULL,
xlab_tr,
ylab_tr,
xlab_den,
ylab_den,
main_den = NULL,
main_tr = NULL,
lwd_den = 1,
lwd_pr = 1,
lty_den = 1,
lty_pr = 1,
col_den,
col_pr,
col_txt,
sz_txt = 1.2,
sz_ax = 1,
sz_ax_txt = 1,
sz_tick_txt = 1,
sz_main_txt = 1.2,
pos_tick_x_tr = NULL,
pos_tick_y_tr = NULL,
pos_tick_x_den = NULL,
pos_tick_y_den = NULL
)

```

Arguments

object	Object containing MCMC output. See DETAILS below.
params	Character string (or vector of character strings) denoting parameters of interest. Default 'all' returns chains for all parameters.
excl	Character string (or vector of character strings) denoting parameters to exclude. Used in conjunction with params argument to select parameters of interest.
ISB	Ignore Square Brackets (ISB). Logical specifying whether square brackets should be ignored in the params and excl arguments. If TRUE, square brackets are ignored. If FALSE, square brackets are not ignored. This allows partial names to

be used when specifying parameters of interest. Use `exact` argument to specify whether input from `params` and `excl` arguments should be matched exactly.

<code>exact</code>	Logical specifying whether input from <code>params</code> and <code>excl</code> arguments should be matched exactly (after ignoring square brackets if <code>ISB = FALSE</code>). If <code>TRUE</code> , input from <code>params</code> and <code>excl</code> are matched exactly (after taking <code>ISB</code> argument into account). If <code>FALSE</code> , input from <code>params</code> and <code>excl</code> are matched using regular expression format (after taking <code>ISB</code> argument into account).
<code>iter</code>	Number of iterations to plot for trace and density plots. The default value is 5000, meaning the last 5000 iterations of the chain will be plotted.
<code>gvals</code>	Vector containing generating values if simulated data was used to fit model. These values will be plotted as vertical lines on the density plots to compare posterior distributions with the true parameter values used to generate the data. No line will be apparent if the generating value is outside the plotted range of the posterior distribution.
<code>priors</code>	Matrix containing random draws from prior distributions corresponding to parameters of interest. If specified, <code>priors</code> are plotted along with posterior density plots. Percent overlap between prior and posterior (PPO) is also calculated and displayed on each plot. Each column of the matrix represents a prior for a different parameter. Parameters are plotted alphabetically - <code>priors</code> should be sorted accordingly. If <code>priors</code> contains only one prior and more than one parameter is specified for the <code>params</code> argument, this prior will be used for all parameters. The number of draws for each prior should equal the number of iterations specified by <code>iter</code> (or total draws if less than <code>iter</code>) times the number of chains, though the function will automatically adjust if more or fewer iterations are specified. See DETAILS below.
<code>post_zm</code>	Logical - if <code>post_zm = FALSE</code> x- and y-limits of density plots are scaled so that both the prior and posterior can be visualized on a single density plot (rather than zoomed on the posterior).
<code>PPO_out</code>	Logical - if <code>PPO_out = TRUE</code> percent overlap between prior and posterior (PPO) will be output to a dataframe.
<code>Rhat</code>	Logical - if <code>Rhat = TRUE</code> potential scale reduction factor (Rhat) for each parameter is plotted on the trace plots.
<code>n.eff</code>	Logical - if <code>n.eff = TRUE</code> number of effective samples for each parameter is plotted on the trace plots.
<code>ind</code>	Logical - if <code>ind = TRUE</code> , separate density lines will be plotted for each chain. If <code>ind = FALSE</code> , one density line will be plotted for all chains.
<code>pdf</code>	Logical - if <code>pdf = TRUE</code> plots will be exported to a pdf.
<code>plot</code>	Logical - if <code>plot = FALSE</code> no plot will be output. Designed to be used in conjunction with <code>PPO_out = TRUE</code> , to calculate PPO without displaying plot output.
<code>open_pdf</code>	Logical - if <code>open_pdf = TRUE</code> pdf will open in viewer after being generated.
<code>filename</code>	Name of pdf file to be printed. Default is 'MCMCtrace'.
<code>wd</code>	Working directory for pdf output. Default is current directory.
<code>type</code>	Type of plot to be output. 'both' outputs both trace and density plots, 'trace' outputs only trace plots, and 'density' outputs only density plots.

<code>ylim</code>	Vector of length two specifying limits for y-axis on density plots only. If specified, overrides argument <code>post_zm</code> .
<code>xlim</code>	Vector of length two specifying limits for x-axis on density plots only. If specified, overrides argument <code>post_zm</code> .
<code>xlab_tr</code>	Character string specifying label for x-axis on trace plots.
<code>ylab_tr</code>	Character string specifying label for x-axis on trace plots.
<code>xlab_den</code>	Character string specifying label for x-axis on density plots.
<code>ylab_den</code>	Character string specifying label for x-axis on density plots.
<code>main_den</code>	Character string (or vector of character strings if plotting > 1 parameter) specifying title(s) of density plot(s).
<code>main_tr</code>	Character string (or vector of character strings if plotting > 1 parameter) specifying title(s) of trace plot(s).
<code>lwd_den</code>	Number specifying thickness of density line on density plots.
<code>lwd_pr</code>	Number specifying thickness of prior line on density plots.
<code>lty_den</code>	Number specifying the line type for the density line on density plots.
<code>lty_pr</code>	Number specifying the line type for the prior line on density plots.
<code>col_den</code>	Character string specifying color of density line on density plots. Does not specify color if <code>ind = TRUE</code> .
<code>col_pr</code>	Character string specifying color of prior line on density plots.
<code>col_txt</code>	Character string specifying color of text (denoting PPO) on plot when value specified for <code>priors</code> . If <code>NULL</code> is specified, no text will be plot.
<code>sz_txt</code>	Number specifying size of text (denoting PPO) when value specified for <code>priors</code> . If <code>NULL</code> is specified, no text will be plot.
<code>sz_ax</code>	Number specifying thickness of axes and ticks.
<code>sz_ax_txt</code>	Number specifying size of text for axes labels.
<code>sz_tick_txt</code>	Number specifying size of text for tick labels on axis.
<code>sz_main_txt</code>	Number specifying size of text for main title.
<code>pos_tick_x_tr</code>	Numeric vector specifying where ticks on x-axis should be placed for trace plots.
<code>pos_tick_y_tr</code>	Numeric vector specifying where ticks on y-axis should be placed for trace plots.
<code>pos_tick_x_den</code>	Numeric vector specifying where ticks on x-axis should be placed for density plots.
<code>pos_tick_y_den</code>	Numeric vector specifying where ticks on y-axis should be placed for density plots.

Details

object argument can be a `stanfit` object (`rstan` package), a `stanreg` object (`rstanarm` package), a `brmsfit` object (`brms` package), an `mcmc.list` object (`coda` and `rjags` packages), `mcmc` object (`coda` and `nimble` packages), `list` object (`nimble` package), an `R2jags` model object (`R2jags` package), a `jagsUI` model object (`jagsUI` package), or a matrix containing MCMC chains (each column representing MCMC output for a single parameter, rows representing iterations in the chain). The function automatically detects the object type and proceeds accordingly.

Matrices for the priors argument can be generated using commands such as `rnorm`, `rgamma`, `runif`, etc. Distributions not supported by base R can be generated by using the appropriate packages. It is important to note that some discrepancies between MCMC samplers and R may exist regarding the parameterization of distributions - one example of this is the use of precision in JAGS but standard deviation in R for the 'second parameter' of the normal distribution. If the number of draws for each prior distribution is greater than the total number used for the density plot (`iter` times the number of chains), the function will use a subset of the prior draws. If the number of draws for each prior distribution is less than the total number used for the density plot, the function will resample (with replacement) from the prior to obtain the appropriate number of draws.

Examples

```
#Load data
data(MCMC_data)

#Traceplots for all 'beta' parameters - pdf is generated by default
MCMCtrace(MCMC_data, params = 'beta', pdf = FALSE)

#Traceplots (individual density lines for each chain) just for 'beta[1]'
MCMCtrace(MCMC_data, params = 'beta[1]',
          ISB = FALSE, exact = TRUE, ind = TRUE, pdf = FALSE)

#Plot prior on top of posterior, calculate prior/posterior overlap (PPO)
#just for 'beta[1]'
#Add Rhat and n.eff values to density plots
PR <- rnorm(15000, 0, 32)
MCMCtrace(MCMC_data, params = 'beta[1]', ISB = FALSE, exact = TRUE,
          priors = PR, pdf = FALSE, Rhat = TRUE, n.eff = TRUE)

#Output PPO to R object without plotting trace plots
PR <- rnorm(15000, 0, 32)
PPO <- MCMCtrace(MCMC_data, params = 'beta[1]', ISB = FALSE, exact = TRUE,
                priors = PR, plot = FALSE, PPO_out = TRUE)
```

MCMC_data

Simulated MCMC output data

Description

Sample MCMC output containing 12 parameters - `alpha[1]`, ... , `alpha[6]`, `beta[1]`, ... , `beta[6]`.

Usage

```
MCMC_data
```

Format

`mcmc.list` object with 3 chains for each parameter, 6000 iterations for each chain.

`MCMC_data2`*Simulated MCMC output data - #2*

Description

Sample MCMC output containing 12 parameters - `alpha[1]`, ... , `alpha[6]`, `beta[1]`, ... , `beta[6]`.

Usage

```
MCMC_data2
```

Format

`mcmc.list` object with 3 chains for each parameter, 6000 iterations for each chain.

Index

* datasets

MCMC_data, [20](#)
MCMC_data2, [21](#)

MCMC_data, [20](#)
MCMC_data2, [21](#)
MCMCchains, [3](#)
MCMCdiag, [5](#)
MCMCplot, [7](#)
MCMCpstr, [12](#)
MCMCsummary, [6](#), [13](#)
MCMCtrace, [16](#)
MCMCvis (MCMCvis-package), [2](#)
MCMCvis-package, [2](#)