Package 'MCMCvis'

March 25, 2020

Type Package

Title Tools to Visualize, Manipulate, and Summarize MCMC Output

Version 0.14.0

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 JAGS, Stan, or other MCMC samplers.

License GPL-3

URL http://github.com/caseyyoungflesh/MCMCvis

BugReports http://github.com/caseyyoungflesh/MCMCvis/issues

LazyData TRUE

Imports coda, rstan, methods, overlapping

Depends R (>= 3.3.0)

RoxygenNote 7.1.0

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Casey Youngflesh [aut, cre] (<https://orcid.org/0000-0001-6343-3311>),
Christian Che-Castaldo [aut] (<https://orcid.org/0000-0002-7670-2178>),
Tyler Hardy [ctb]

Repository CRAN

Date/Publication 2020-03-25 05:20:02 UTC

R topics documented:

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,
  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 - input from params and excl are otherwise matched exactly. If FALSE, square brackets are not ignored - input from params and excl are matched using regular expression format. This allows partial names to be used when specifying parameters of interest.
mcmc.list	Logical specifying whether to return an mcmc.list. If TRUE, an mcmc.list object is returned, rather than a matrix.

chain_num   Numeric - specifies posterior chain number. When a value is specified, poste-
                 rior 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 stanreg object (rstanarm package),
a brmsfit object (brms package), an mcmc.list object (coda 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]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\'
ex3 <- MCMCchains(MCMC_data, params = c('beta\\[1\\]', 'beta\\[4\\]', 'alpha\\[3\\]'), ISB = FALSE)
apply(ex3, 2, sd)
```

MCMCplot   *Caterpillar plots of posterior distributions from MCMC output*

Description

Visualize posterior distributions from MCMC output for specific parameters of interest using cater-
pillar 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",
  excl = NULL,
  ISB = 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.
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 ig-

	nored - input from `params` and `excl` are otherwise matched exactly. If `FALSE`, square brackets are not ignored - input from `params` and `excl` are matched using grep, which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.
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 50 % and 95 % credible intervals 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).
xlim	Numerical vector of length 2, indicating range of x-axis. Only applicable if `horiz = TRUE`.
ylim	Numerical vector of length 2, indicating range of y-axis. Only applicable if `horiz = FALSE`.
xlab	Character string labeling x-axis. Only applicable if `horiz = TRUE`. Default label is 'Parameter Estimate'. Option `NULL` will return plot with no label on x-axis.
ylab	Character string labeling y-axis. Only applicable if `horiz = FALSE`. Default label is 'Parameter Estimate'. Option `NULL` will return plot with no label on y-axis.
main	Character string indicating title of plot.
labels	Character string (or vector of character strings if plotting > 1 parameter) labeling parameter estimates along y-axis (if `horiz = FALSE`) or x-axis (if `horiz = TRUE`). Default option will use parameter names from `object`. Option `NULL` will return plot with no labels on axis.

guide_lines	Logical specifying whether to plot reference lines for each parameter in order to better visualize which parameter names correspond to each posterior.
guide_axis	Logical specifying whether a second axis should be plotted (x-axis if HORIZ = TRUE, y-axis if HORIZ = FALSE) to help interpret values on plot.
sz_labels	Number specifying size of text for parameter labels on axis.
sz_med	Number specifying size of points represents posterior medians.
sz_thick	Number specifying thickness of 50 percent CI line (thicker line).
sz_thin	Number specifying thickness of 95 percent CI line (thinner line).
sz_ax	Number specifying thickness of axis and ticks.
sz_ax_txt	Number specifying size of text for axis label.
sz_tick_txt	Number specifying size of text for tick labels on axis.
sz_main_txt	Number specifying size of text for main title.
pos_tick	Numeric vector specifying where ticks on axis should be placed.
mar	Numerical vector of length 4 specifying plot margins - (BOTTOM, LEFT, TOP, RIGHT). Changes to the margin should be made within the function rather than using the par call.
	Default is c(5.1, 4.1, 4.1, 2.1) - the R plot default.

Details

Points represent posterior medians. Parameters where 50% credible intervals overlap 0 (or other specified value) are indicated by 'open' circles. Parameters where 50 percent credible intervals DO NOT overlap 0 AND 95 percent credible intervals DO overlap 0 (or other specified value) are indicated by 'closed' gray circles. Parameters where 95 percent credible intervals DO NOT overlap 0 (or other specified value) are indicated by 'closed' black circles. Thick lines represent 50 percent credible intervals while thin lines represent 95 % credible intervals. ref_ovl = TRUE can be used to enable this feature.

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 stanreg object (rstanarm package), a brmsfit object (brms package), an mcmc.list object (coda 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[1]', 'beta[4]', and 'alpha[3]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\'
MCMCplot(MCMC_data, params = c('beta\\[1\\]', 'beta\\[4\\]', 'alpha\\[3\\]'), ISB = FALSE)

#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,
  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 re-
                turned 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 `params` and `excl` arguments. If TRUE, square brackets are ig-
                nored - input from `params` and `excl` are otherwise matched exactly. If FALSE,
                square brackets are not ignored - input from `params` and `excl` are matched using
                grep, which can take arguments in regular expression format. This allows partial
                names to be used when specifying parameters of interest.

func   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. Func-
                tions that produce output with dimensionality greater than 1 are not permitted.
                `func` is ignored when `type = 'chains'`.

type   Character string specifying whether to return summary information (calculated
                based on `func` argument) or posterior chains. Valid options are `'summary'` and
                `'chains'`. When `type = 'chains'`, the `'func'` argument is ignored. When
                `type = 'chains'`, posterior chains are concatenated and stored in the last di-
                mension in the array for each element (parameter) of the list.

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` 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,
  probs = c(0.025, 0.5, 0.975),
  hpd_prob = 0.95,
  HPD = 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 - input from `params` and `excl` are otherwise matched exactly. If `FALSE`, square brackets are not ignored - input from `params` and `excl` are matched using grep, which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.
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.
digits	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.
round	Number of decimal places to round to for posterior summary. Cannot be used in conjunction with digits argument. Note that Rhat is always rounded to 2 decimal places.
Rhat	Logical specifying whether to calculate and display the potential scale reduction statistic (Rhat). Values near 1 suggest convergence (Brooks and Gelman 1998). Rhat = FALSE will prevent display of this column in summary output. Specifying Rhat = FALSE, may increase function speed for very large mcmc.list objects.
n.eff	Logical specifying whether to calculate and display the number of effective samples for each parameter. n.eff = FALSE will prevent display of this column in summary output. Specifying n.eff = FALSE, may increase function speed for very large mcmc.list objects. Default is n.eff = TRUE.
func	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).
func_name	Character string (or vector of character strings) specifying labels for output from func argument. If func_name is not specified, columns with func 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 stanreg object (rstanarm package), a brmsfit object (brms package), an mcmc.list object (coda 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 objects, the potential scale reduction statistic 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 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` objects, the number of effective samples is calculated using the `effectiveSize` function in the coda package. For `stanfit` (as well as `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).

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. http://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]'
#'params' takes regular expressions when ISB = FALSE, square brackets must be escaped with '\'
MCMCsummary(MCMC_data, params = c('beta\\[1\\]', 'beta\\[4\\]', 'alpha\\[3\\]'), ISB = FALSE)
```

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,
  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 - input from `params` and `excl` are otherwise matched exactly. If `FALSE`, square brackets are not ignored - input from `params` and `excl` are matched using grep, which can take arguments in regular expression format. This allows partial names to be used when specifying parameters of interest.

iter	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.
gvals	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.
priors	Matrix containing random draws from prior distributions corresponding to parameters of interest. If specified, priors 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 - priors should be sorted accordingly. If `priors` contains only one prior and more than one parameter is specified for the `params` argument, this prior will be used for all parameters. The number of draws for each prior should equal the number of iterations specified by `iter` (or total draws if less than `iter`) times the number of chains, though the function will automatically adjust if more or fewer iterations are specified. See DETAILS below.
post_zm	Logical - if `post_zm = FALSE` 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).
PPO_out	Logical - if `PPO_out = TRUE` percent overlap between prior and posterior (PPO) will be output to a dataframe.
Rhat	Logical - if `Rhat = TRUE` potential scale reduction factor (Rhat) for each parameter is plotted on the trace plots.
n.eff	Logical - if `n.eff = TRUE` number of effective samples for each parameter is plotted on the trace plots.
ind	Logical - if `ind = TRUE`, separate density lines will be plotted for each chain. If `ind= FALSE`, one density line will be plotted for all chains.
pdf	Logical - if `pdf = TRUE` plots will be exported to a pdf.
plot	Logical - if `plot = FALSE` no plot will be output. Designed to be used in conjunction with `PPO_out = TRUE`, to calculate PPO without displaying plot output.
open_pdf	Logical - if `open_pdf = TRUE` pdf will open in viewer after being generated.
filename	Name of pdf file to be printed. Default is 'MCMCtrace'.
wd	Working directory for pdf output. Default is current directory.
type	Type of plot to be output. `'both'` outputs both trace and density plots, `'trace'` outputs only trace plots, and `'density'` outputs only density plots.
ylim	Vector of length two specifying limits for y-axis on density plots only. If specified, overrides argument `post_zm`.
xlim	Vector of length two specifying limits for x-axis on density plots only. If specified, overrides argument `post_zm`.
xlab_tr	Character string specifying label for x-axis on trace plots.
ylab_tr	Character string specifying label for x-axis on trace plots.

xlab_den	Character string specifying label for x-axis on density plots.
ylab_den	Character string specifying label for x-axis on density plots.
main_den	Character string (or vector of character strings if plotting > 1 parameter) specifying title(s) of density plot(s).
main_tr	Character string (or vector of character strings if plotting > 1 parameter) specifying title(s) of trace plot(s).
lwd_den	Number specifying thickness of density line on density plots.
lwd_pr	Number specifying thickness of prior line on density plots.
lty_den	Number specifying the line type for the density line on density plots.
lty_pr	Number specifying the line type for the prior line on density plots.
col_den	Character string specifying color of density line on density plots. Does not specify color if ind = TRUE.
col_pr	Character string specifying color of prior line on density plots.
col_txt	Character string specifying color of text (denoting PPO) on plot when value specified for priors. If NULL is specified, no text will be plot.
sz_txt	Number specifying size of text (denoting PPO) when value specified for priors. If NULL is specified, no text will be plot.
sz_ax	Number specifying thickness of axes and ticks.
sz_ax_txt	Number specifying size of text for axes labels.
sz_tick_txt	Number specifying size of text for tick labels on axis.
sz_main_txt	Number specifying size of text for main title.
pos_tick_x_tr	Numeric vector specifying where ticks on x-axis should be placed for trace plots.
pos_tick_y_tr	Numeric vector specifying where ticks on y-axis should be placed for trace plots.
pos_tick_x_den	Numeric vector specifying where ticks on x-axis should be placed for density plots.
pos_tick_y_den	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 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]'
#'params' takes regular expressions when ISB = FALSE, square brackets
#must be escaped with '\'
MCMCtrace(MCMC_data, params = 'beta\\[1\\]',
          ISB = FALSE, 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
#'params' takes regular expressions when ISB = FALSE, square brackets must
#be escaped with '\'
PR <- rnorm(15000, 0, 32)
MCMCtrace(MCMC_data, params = 'beta\\[1\\]', ISB = FALSE, 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,
                 priors = PR, plot = FALSE, PPO_out = TRUE)
```

MCMCvis   *The 'MCMCvis' package*

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)

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>

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