

# Package ‘ggmcmc’

July 2, 2014

**Title** Graphical tools for analyzing Markov Chain Monte Carlo simulations from Bayesian inference.

**Description** ggmcmc is a tool for assessing and diagnosing convergence of Markov Chain Monte Carlo simulations, as well as for graphically display results from full MCMC analysis. The package also facilitates the graphical interpretation of models by providing flexible functions to plot the results against observed variables.

**Version** 0.5.1

**Maintainer** Xavier Fernández i Marín <xavier.fim@gmail.com>

**Author** Xavier Fernández i Marín <xavier.fim@gmail.com>

**Depends** plyr, reshape2, ggplot2

**Suggests** foreach, rjags, coda, rstan

**License** GPL-2

**URL** <http://xavier-fim.net/packages/ggmcmc>

**Encoding** UTF-8

**Collate** 'ggmcmc.R' 'ggs.R' 'ggs\_autocorrelation.R' 'ggs\_compare\_partial.R' 'ggs\_crosscorrelation.R' 'ggs\_density.R' 'ggs\_histogram.R' 'ggs\_running.R' 'ggs\_traceplot.R' 'data.R' 'help.R' 'functions.R' 'ggs\_Rhat.R' 'ggs\_geweke.R' 'ggs\_caterpillar.R' 'ggs\_separation.R' 'globals.R' 'ggs\_ppmean.R' 'ggs\_ppsd.R' 'ggs\_rocplot.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-10-03 19:41:00

## R topics documented:

ac	2
calc.bin	3
get_family	3
ggmcmc	4
ggs	5
ggs_autocorrelation	7
ggs_caterpillar	7
ggs_chain	8
ggs_compare_partial	9
ggs_crosscorrelation	10
ggs_density	10
ggs_geweke	11
ggs_histogram	12
ggs_ppmean	13
ggs_ppsd	14
ggs_Rhat	15
ggs_rocplot	15
ggs_running	16
ggs_separation	17
ggs_traceplot	18
gl.unq	19
roc.calc	19
S	20
sde0f	20
<b>Index</b>	<b>21</b>

---

ac	<i>Calculate the autocorrelation of a single chain, for a specified amount of lags</i>
----	----------------------------------------------------------------------------------------

---

### Description

Calculate the autocorrelation of a single chain, for a specified amount of lags.

### Usage

```
ac(x, nLags)
```

### Arguments

x	Vector with a chain of simulated values.
nLags	Numerical value with the maximum number of lags to take into account.

**Details**

Internal function used by [ggs\\_autocorrelation](#).

**Value**

A matrix with the autocorrelations of every chain.

**Examples**

```
# Calculate the autocorrelation of a simple vector
ac(cumsum(rnorm(10))/10, nLags=4)
```

---

calc.bin	<i>Calculate binwidths by parameter, based on the total number of bins</i>
----------	----------------------------------------------------------------------------

---

**Description**

Compute the minimal elements to recreate a histogram manually by defining the total number of bins

**Usage**

```
calc.bin(x, bins = bins)
```

**Arguments**

x	any vector or variable
bins	the number of requested bins

**Value**

A data frame with the x location, the width of the bars and the number of observations at each x location.

---

get_family	<i>Subset a ggs object to get only the parameters with a given regular expression.</i>
------------	----------------------------------------------------------------------------------------

---

**Description**

Internal function used by the graphical functions to get only some of the parameters that follow a given regular expression.

**Usage**

```
get_family(D, family = NA)
```

**Arguments**

D	Data frame with the data arranged and ready to be used by the rest of the ggmcmc functions. The dataframe has four columns, namely: Iteration, Parameter, value and Chain, and seven attributes: nChains, nParameters, nIterations, nBurnin, nThin, description and parallel.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).

**Value**

D Data frame that is a subset of the given D dataset.

---

ggmcmc	<i>Wrapper function that creates a single pdf file with all plots that ggmcmc can produce.</i>
--------	------------------------------------------------------------------------------------------------

---

**Description**

ggmcmc() is simply a wrapper function that generates a pdf file with all the potential plots that the package can produce.

ggmcmc is a tool for assessing and diagnosing convergence of Markov Chain Monte Carlo simulations, as well as for graphically display results from full MCMC analysis. The package also facilitates the graphical interpretation of models by providing flexible functions to plot the results against observed variables.

**Usage**

```
ggmcmc(D, file = "ggmcmc-output.pdf", family = NA,
       plot = NULL, param_page = 5, width = 7, height = 10,
       ...)
```

**Arguments**

D	Data frame with the simulations, previously arranged using <a href="#">ggs</a>
file	Character vector with the name of the file to create. Defaults to "ggmcmc-output.pdf". When NULL, no pdf device is opened or closed. This allows the user to work with an opened pdf (or other) device.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).

plot	character vector containing the names of the desired plots. By default (NULL), ggmcnc() plots ggs_histogram(), ggs_density(), ggs_traceplot(), ggs_running(), ggs_compare_partial(), ggs_autocorrelation(), ggs_crosscorrelation(), ggs_Rhat(), ggs_geweke() and ggs_caterpillar().
param_page	Numerical, number of parameters to plot for each page. Defaults to 5.
width	Width of the pdf display, in inches. Defaults to 7.
height	Height of the pdf display, in inches. Defaults to 10.
...	Other options passed to the pdf device.

### Details

Notice that caterpillar plots are only created when there are multiple parameters within the same family. A family of parameters is considered to be all parameters that have the same name (usually the same greek letter) but different number within square brackets (such as  $\alpha[1]$ ,  $\alpha[2]$ , ...).

### References

<http://xavier-fim.net/packages/ggmcnc>.

### Examples

```
data(samples)
ggmcnc(ggs(S)) # Directly from a coda object
```

---

ggs	<i>Import MCMC samples into a ggs object than can be used by all ggs_* graphical functions.</i>
-----	-------------------------------------------------------------------------------------------------

---

### Description

This function manages MCMC samples from different sources (JAGS, MCMCpack, STAN -both via rstan and via csv files-) and converts them into a data frame. The resulting data frame has four columns (Iteration, Parameter, value, Chain) and seven attributes (nChains, nParameters, nIterations, nBurnin, nThin, description and parallel). The ggs object returned is then used as the input of the ggs\_\* functions to actually plot the different convergence diagnostics.

### Usage

```
ggs(S, family = NA, description = NA, burnin = TRUE,
    par_labels = NA, inc_warmup = FALSE,
    stan_include_auxiliar = FALSE, parallel = FALSE)
```

## Arguments

<code>S</code>	Either a <code>mcmc.list</code> object with samples from JAGS, a <code>mcmc</code> object with samples from MCMCpack, a <code>stanfit</code> object with samples from rstan, or a list with the filenames of csv files generated by stan outside rstan (where the order of the files is assumed to be the order of the chains). <code>ggmcmc</code> guesses what is the original object and tries to import it accordingly. rstan is not expected to be in CRAN soon, and so <code>coda::mcmc</code> is used to extract stan samples instead of the more canonical <code>rstan::extract</code> .
<code>family</code>	Name of the family of parameters to process, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as <code>beta[1]</code> , <code>beta[2]</code> , etc).
<code>description</code>	Character vector giving a short descriptive text that identifies the model.
<code>burnin</code>	Logical or numerical value. When logical and TRUE (the default), the number of samples in the burnin period will be taken into account, if it can be guessed by the extracting process. Otherwise, iterations will start counting from 1. If a numerical vector is given, the user then supplies the length of the burnin period.
<code>par_labels</code>	data frame with two columns. One named "Parameter" with the same names of the parameters of the model. Another named "Label" with the label of the parameter. When missing, the names passed to the model are used for representation. When there is no correspondence between a Parameter and a Label, the original name of the parameter is used. The order of the levels of the original Parameter does not change.
<code>inc_warmup</code>	Logical. When dealing with stanfit objects from rstan, logical value whether the warmup samples are included. Defaults to FALSE.
<code>stan_include_auxiliar</code>	Logical value to include "lp__" parameter in rstan, and "lp__", "treedepth__" and "stepsize__" in stan running without rstan. Defaults to FALSE.
<code>parallel</code>	Logical value for using parallel computing when managing the data frame in other functions. Defaults to FALSE. It needs a registered parallel backend. Currently it has only been extensively tested in doMC.

## Value

D A data frame with the data arranged and ready to be used by the rest of the `ggmcmc` functions. The data frame has four columns, namely: Iteration, Parameter, value and Chain, and seven attributes: `nChains`, `nParameters`, `nIterations`, `nBurnin`, `nThin`, `description` and `parallel`.

## Examples

```
# Assign 'D' to be a data frame suitable for \code{ggmcmc} functions from
# a coda object called S
data(samples)
D <- ggs(S)      # S is a coda object
```

---

`ggs_autocorrelation` *Plot an autocorrelation matrix*

---

### Description

Plot an autocorrelation matrix.

### Usage

```
ggs_autocorrelation(D, family = NA, nLags = 50)
```

### Arguments

<code>D</code>	Data frame with the simulations.
<code>family</code>	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as <code>beta[1]</code> , <code>beta[2]</code> , etc).
<code>nLags</code>	Integer indicating the number of lags of the autocorrelation plot.

### Value

A ggplot object.

### Examples

```
data(samples)
ggs_autocorrelation(ggs(S))
```

---

`ggs_caterpillar` *Caterpillar plot with thick and thin CI*

---

### Description

Caterpillar plots are plotted combining all chains for each parameter.

### Usage

```
ggs_caterpillar(D, family = NA, X = NA,
  thick_ci = c(0.05, 0.95), thin_ci = c(0.025, 0.975),
  line = NA, horizontal = TRUE, model_labels = NULL)
```

**Arguments**

D	Data frame with the simulations or list of data frame with simulations. If a list of data frames with simulations is passed, the names of the models are the names of the objects in the list.
X	data frame with two columns, Parameter and the value for the x location. Parameter must be a character vector with the same names that the parameters in the D object.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).
thick_ci	Vector of length 2 with the quantiles of the thick band for the credible interval
thin_ci	Vector of length 2 with the quantiles of the thin band for the credible interval
line	Numerical value indicating a concrete position, usually used to mark where zero is. By default do not plot any line.
horizontal	Logical. When TRUE (the default), the plot has horizontal lines. When FALSE, the plot is reversed to show vertical lines. Horizontal lines are more appropriate for categorical caterpillar plots, because the x-axis is the only dimension that matters. But for caterpillar plots against another variable, the vertical position is more appropriate.
model_labels	Vector of strings that matches the number of models in the list. It is only used in case of multiple models and when the list of ggs objects given at D is not named. Otherwise, the names in the list are used.

**Value**

A ggplot object.

**Examples**

```
data(samples)
ggs_caterpillar(ggs(S))
ggs_caterpillar(list(A=ggs(S), B=ggs(S))) # silly example duplicating the same model
```

---

ggs\_chain

*Auxiliary function that extracts information from a single chain.*


---

**Description**

Auxiliary function that extracts information from a single chain.

**Usage**

```
ggs_chain(s)
```



**Arguments**

s a single chain to convert into a data frame

**Value**

D data frame with the chain arranged

---

ggs\_compare\_partial *Density plots comparing the distribution of the whole chain with only its last part.*

---

**Description**

Density plots comparing the distribution of the whole chain with only its last part.

**Usage**

```
ggs_compare_partial(D, family = NA, partial = 0.1,  
rug = FALSE)
```

**Arguments**

D Data frame with the simulations

family Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).

partial Percentage of the chain to compare to. Defaults to the last 10 percent.

rug Logical indicating whether a rug must be added to the plot. It is FALSE by default, since in large chains it may use a lot of resources and it is not central to the plot.

**Value**

A ggplot object.

**Examples**

```
data(samples)  
ggs_compare_partial(ggs(S))
```

---

`ggs_crosscorrelation` *Plot the Cross-correlation between-chains*

---

### Description

Plot the Cross-correlation between-chains.

### Usage

```
ggs_crosscorrelation(D, family = NA,  
  absolute_scale = TRUE)
```

### Arguments

`D` Data frame with the simulations.

`family` Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as `beta[1]`, `beta[2]`, etc).

`absolute_scale` Logical. When `TRUE` (the default), the scale of the colour diverges between perfect inverse correlation (-1) to perfect correlation (1), whereas when `FALSE`, the scale is relative to the minimum and maximum cross-correlations observed.

### Value

a `ggplot` object.

### Examples

```
data(samples)  
ggs_crosscorrelation(ggs(S))
```

---

`ggs_density` *Density plots of the chains*

---

### Description

Density plots with the parameter distribution. For multiple chains, use colours to differentiate the distributions.

### Usage

```
ggs_density(D, family = NA, rug = FALSE)
```

**Arguments**

D	Data frame with the simulations.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).
rug	Logical indicating whether a rug must be added to the plot. It is FALSE by default, since in large chains it may use lot of resources and it is not central to the plot.

**Value**

A ggplot object.

**Examples**

```
data(samples)
ggs_density(ggs(S))
```

---

ggs\_geweke

*Dotplot of the Geweke diagnostic, the standard Z-score*

---

**Description**

Dotplot of Geweke diagnostic.

**Usage**

```
ggs_geweke(D, family = NA, frac1 = 0.1, frac2 = 0.5,
  shadow_limit = TRUE)
```

**Arguments**

D	data frame with the simulations.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).
frac1	Numeric, proportion of the first part of the chains selected. Defaults to 0.1.
frac2	Numeric, proportion of the last part of the chains selected. Defaults to 0.5.
shadow_limit,	logical. When TRUE (the default), a shadowed area between -2 and +2 is drawn.

**Value**

A ggplot object.

**Examples**

```
data(samples)
ggs_geweke(ggs(S))
```

---

ggs\_histogram

*Histograms of the paramters.*

---

**Description**

Plot a histogram of each of the parameters. Histograms are plotted combining all chains for each parameter.

**Usage**

```
ggs_histogram(D, family = NA, bins = 30)
```

**Arguments**

D	Data frame whith the simulations.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).
bins	integer indicating the total number of bins in which to divide the histogram. Defaults to 30, which is the same as geom_histogram()

**Value**

A ggplot object.

**Examples**

```
data(samples)
ggs_histogram(ggs(S))
```

---

ggs_ppmean	<i>Posterior predictive plot comparing the outcome vs the posterior means.</i>
------------	--------------------------------------------------------------------------------

---

## Description

Histogram with the distribution of the predicted posterior means, compared with the mean of the observed outcome.

## Usage

```
ggs_ppmean(D, outcome, family = NA, bins = 30)
```

## Arguments

D	Data frame with the simulations. Notice that only the posterior outcomes are needed, and so either the <code>ggs()</code> call limits the parameters to the outcomes or the user provides a family of parameters to limit it.
outcome	vector (or matrix or array) containing the observed outcome variable. Currently only a vector is supported.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as <code>beta[1]</code> , <code>beta[2]</code> , etc).
bins	integer indicating the total number of bins in which to divide the histogram. Defaults to 30, which is the same as <code>geom_histogram()</code>

## Value

A ggplot object.

## Examples

```
## Not run:  
data(samples)  
ggs_ppmean(ggs(S, family="y"), outcome="y")  
  
## End(Not run)
```

---

ggs_ppsd	<i>Posterior predictive plot comparing the outcome vs the posterior standard deviations.</i>
----------	----------------------------------------------------------------------------------------------

---

## Description

Histogram with the distribution of the predicted posterior standard deviations, compared with the standard deviations of the observed outcome.

## Usage

```
ggs_ppsd(D, outcome, family = NA, bins = 30)
```

## Arguments

D	Data frame with the simulations. Notice that only the posterior outcomes are needed, and so either the <code>ggs()</code> call limits the parameters to the outcomes or the user provides a family of parameters to limit it.
outcome	vector (or matrix or array) containing the observed outcome variable. Currently only a vector is supported.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as <code>beta[1]</code> , <code>beta[2]</code> , etc).
bins	integer indicating the total number of bins in which to divide the histogram. Defaults to 30, which is the same as <code>geom_histogram()</code>

## Value

A ggplot object.

## Examples

```
## Not run:  
data(samples)  
ggs_ppsd(ggs(S, family="y"), outcome="y")  
  
## End(Not run)
```

ggs\_Rhat

*Dotplot of Potential Scale Reduction Factor (Rhat)***Description**

Plot a dotplot of Potential Scale Reduction Factor (Rhat), proposed by Gelman and Rubin (1992). The version from the second edition of Bayesian Data Analysis (Gelman, Carlin, Stein and Rubin) is used.

**Usage**

```
ggs_Rhat(D, family = NA, scaling = 1.5)
```

**Arguments**

D	Data frame with the simulations
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).
scaling	Value of the upper limit for the x-axis. By default, it is 1.5, to help contextualization of the convergence. When 0 or NA, the axis are not scaled.

**Details**

Notice that at least two chains are required.

**Value**

A ggplot object.

**Examples**

```
data(samples)
ggs_Rhat(ggs(S))
```

ggs\_rocplot

*Receiver-Operator Characteristic (ROC) plot for models with binary outcomes***Description**

Receiver-Operator Characteristic (ROC) plot for models with binary outcomes

**Usage**

```
ggs_rocplot(D, outcome, fully_bayesian = FALSE)
```

**Arguments**

**D** Data frame with the simulations. Notice that only the posterior outcomes are needed, and so either the previous call to `ggs()` should have limited the family of parameters to pass to the predicted outcomes.

**outcome** vector (or matrix or array) containing the observed outcome variable. Currently only a vector is supported.

**fully\_bayesian** logical, false by default. When not fully Bayesian, it uses the median of the predictions for each observation by iteration. When TRUE the function plots as many ROC curves as iterations. It uses a lot of CPU and needs more memory. Use it with caution.

**Value**

A ggplot object

**Examples**

```
## Not run:
ggs_rocplot(S, outcome=y)

## End(Not run)
```

---

ggs\_running

*Running means of the chains*

---

**Description**

Running means of the chains.

**Usage**

```
ggs_running(D, family = NA, original_burnin = TRUE,
            original_thin = TRUE)
```

**Arguments**

**D** Data frame with the simulations.

**family** Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as `beta[1]`, `beta[2]`, etc).



original_burnin	Logical. When TRUE (the default), start the iteration counter in the x-axis at the end of the burnin period.
original_thin	Logical. When TRUE (the default), take into account the thinning interval in the x-axis.

**Value**

A ggplot object.

**Examples**

```
data(samples)
ggs_running(ggs(S))
```

---

ggs_separation	<i>Separation plot for models with binary response variables</i>
----------------	------------------------------------------------------------------

---

**Description**

Separation plot for models with binary response variables

**Usage**

```
ggs_separation(D, outcome, fully_bayesian = FALSE,
              minimalist = FALSE)
```

**Arguments**

D	Data frame which the simulations. Notice that only the posterior outcomes are needed, and so either the previous call to ggs() should have limited the family of parameters to pass to the predicted outcomes.
outcome	vector (or matrix or array) containing the observed outcome variable. Currently only a vector is supported.
fully_bayesian	logical, FALSE by default. Currently not implemented
minimalist	logical, FALSE by default. It returns a minimalistic version of the figure with the bare minimum elements, suitable for being used inline as suggested by Greenhill, Ward and Sacks citing Tufte.

**Value**

A ggplot object

**References**

Greenhill, Ward and Sacks (2011): The separation plot: a new visual method for evaluating the fit of binary models. American Journal of Political Science, vol 55, number 4, pg 991-1002.

## Examples

```
## Not run:  
ggs_separation(S, outcome=y)  
  
## End(Not run)
```

---

ggs_traceplot	<i>Traceplot of the chains</i>
---------------	--------------------------------

---

## Description

Traceplot with the time series of the chains.

## Usage

```
ggs_traceplot(D, family = NA, original_burnin = TRUE,  
              original_thin = TRUE)
```

## Arguments

D	Data frame with the simulations.
family	Name of the family of parameters to plot, as given by a character vector or a regular expression. A family of parameters is considered to be any group of parameters with the same name but different numerical value between square brackets (as beta[1], beta[2], etc).
original_burnin	Logical. When TRUE (the default) start the Iteration counter in the x-axis at the end of the burnin period.
original_thin	Logical. When TRUE (the default) take into account the thinning interval in the x-axis.

## Value

A ggplot object.

## Examples

```
data(samples)  
ggs_traceplot(ggs(S, parallel=FALSE))
```

---

gl.unq	<i>Generate a factor with unequal number of repetitions</i>
--------	-------------------------------------------------------------

---

**Description**

Generate a factor with levels of unequal length

**Usage**

```
gl.unq(n, k, labels = 1:n)
```

**Arguments**

n	number of levels
k	number of repetitions
labels	optional vector of labels

**Value**

A factor

---

roc.calc	<i>Calculate the ROC curve for a set of observed outcomes and predicted probabilities</i>
----------	-------------------------------------------------------------------------------------------

---

**Description**

Internal function used by [ggs\\_autocorrelation](#).

**Usage**

```
roc.calc(R)
```

**Arguments**

R	data frame with the 'value' (predicted probability) and the observed 'Outcome'.
---	---------------------------------------------------------------------------------

**Value**

A data frame with the Sensitivity and the Specificity.

---

S	<i>Simulations of the parameters of a simple linear regression with fake data.</i>
---	------------------------------------------------------------------------------------

---

**Description**

A coda object containing simulated values from the posterior distribution of the intercept, slope and error of a simple linear regression with fake data ( $y = \text{beta}[0] + \text{beta}[1] * X + \text{sigma}$ ) . The purpose of the dataset is only to show the possibilities of the ggcmc package.

**Format**

A coda object with 2 chains of 1000 iterations each one for 3 parameters

---

sde0f	<i>Spectral Density Estimate at Zero Frequency</i>
-------	----------------------------------------------------

---

**Description**

Compute the Spectral Density Estimate at Zero Frequency for a given chain.

**Usage**

```
sde0f(x)
```

**Arguments**

x                    A time series

**Value**

A vector with the spectral density estimate at zero frequency

# Index

\*Topic **datasets**

S, [20](#)

ac, [2](#)

calc.bin, [3](#)

get\_family, [3](#)

ggmcmc, [4](#)

ggmcmc-package (ggmcmc), [4](#)

ggs, [4](#), [5](#)

ggs\_autocorrelation, [3](#), [7](#), [19](#)

ggs\_caterpillar, [7](#)

ggs\_chain, [8](#)

ggs\_compare\_partial, [9](#)

ggs\_crosscorrelation, [10](#)

ggs\_density, [10](#)

ggs\_geweke, [11](#)

ggs\_histogram, [12](#)

ggs\_ppmean, [13](#)

ggs\_ppsd, [14](#)

ggs\_Rhat, [15](#)

ggs\_rocplot, [15](#)

ggs\_running, [16](#)

ggs\_separation, [17](#)

ggs\_traceplot, [18](#)

gl.unq, [19](#)

roc.calc, [19](#)

S, [20](#)

sde0f, [20](#)