

# Package ‘scapeMCMC’

January 2, 2012

**Version** 1.1-3

**Date** 2011-09-29

**Title** MCMC Diagnostic Plots

**Author** Arni Magnusson and Ian Stewart

**Maintainer** Arni Magnusson <arnima@hafro.is>

**Depends** coda, lattice

**Imports** gplots

**Suggests** gdata

**LazyData** yes

**Description** Markov chain Monte Carlo diagnostic plots. The purpose of the package is to combine existing tools from the ‘coda’ and ‘lattice’ packages, and make it easy to adjust graphical details. It can be useful for anyone using MCMC analysis, regardless of the application.

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2011-09-30 05:52:15

## R topics documented:

scapeMCMC-package . . . . .	2
importMCMC . . . . .	3
importProj . . . . .	4
plotAuto . . . . .	6
plotCumu . . . . .	7
plotDens . . . . .	8
plotQuant . . . . .	10
plotSplom . . . . .	12
plotTrace . . . . .	13
xmcmc . . . . .	15
xproj . . . . .	16

---

scapeMCMC-package      *MCMC Diagnostic Plots*

---

### Description

Markov chain Monte Carlo diagnostic plots. The purpose of the package is to combine existing tools from the **cod**a and **latt**ice packages, and make it easy to adjust graphical details. It can be useful for anyone using MCMC analysis, regardless of the application.

### Details

*Import Coleraine MCMC results:*

<code>importMCMC</code>	traces of likelihoods, parameters, biomass and recruitment
<code>importProj</code>	future projections of biomass and catch

*Diagnostic plots:*

<code>plotTrace</code>	trends
<code>plotAuto</code>	thinning
<code>plotCumu</code>	convergence
<code>plotSplom</code>	confounding of parameters

*Posterior plots:*

<code>plotDens</code>	posterior(s)
<code>plotQuant</code>	multiple posteriors on a common y axis

*Examples:*

<code>xmcmc</code> , <code>xproj</code>	MCMC results and projections
---	------------------------------

### Note

`browseVignettes()` shows the vignettes. The main vignette ‘scapeMCMC/doc/dsc.pdf’ covers **scapeMCMC** in Section 2.4.

The plot functions assume that MCMC results are stored either as a plain vector (single chain) or in named columns (multiple chains). It should be easy for users to arrange their MCMC results in this way. The examples demonstrate how several data frames can be stored together in nested lists.

The functions `Args` and `l1` (package **gdata**) can be useful for browsing unwieldy functions and objects.

### Author(s)

Arni Magnusson and Ian Stewart.

## References

Magnusson, A. (2005) *R goes fishing: Analyzing fisheries data using AD Model Builder and R*. Proceedings of the 5th International Workshop on Distributed Statistical Computing. Available at <http://www.hafro.is/~arnima/uw/s/pdf/dsc.pdf>.

## See Also

The **coda** package is a suite of diagnostic functions and plots for MCMC analysis, many of which are used in **scapeMCMC**.

---

importMCMC	<i>Import Coleraine MCMC Results</i>
------------	--------------------------------------

---

## Description

Import Coleraine MCMC traces for likelihoods, parameters, spawning biomass, and recruitment.

## Usage

```
importMCMC(dir, info="", coda=FALSE, quiet=TRUE, pretty.labels=FALSE,
           l.choose=NULL, p.choose=NULL)
```

## Arguments

<code>dir</code>	directory containing the files 'mcmclike.out', 'params.pst', 'spawbiom.pst' and 'recruits.pst'.
<code>info</code>	optional string containing information to store with MCMC results.
<code>coda</code>	whether data frames should be coerced to class <code>mcmc</code> using the <b>coda</b> package.
<code>quiet</code>	whether to report progress while parsing files in directory.
<code>pretty.labels</code>	whether likelihood and parameter columns should be renamed
<code>l.choose</code>	vector of strings, indicating which likelihood components to import, or NULL to import all.
<code>p.choose</code>	vector of strings, indicating which parameters to import, or NULL to import all.

## Value

A list containing:

L	likelihoods
P	parameters
B	biomass by year
R	recruitment by year

as data frames, or `mcmc` objects if `coda=TRUE`.

**Note**

The example dataset `xmcmc` was generated using `importMCMC`.

The functions `ll` (package **gdata**) and `head` are recommended for browsing MCMC results, e.g. `ll(xmcmc)`; `ll(xmcmc$P)`; `head(xmcmc$P)`.

**References**

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model*. User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at <http://fish.washington.edu/research/coleraine/coleraine.pdf>.

**See Also**

`read.table`, `readLines`, and `scan` import any data.

`importMCMC` and `importProj` import Coleraine MCMC results.

`xmcmc` was created using `importMCMC`.

`scapeMCMC-package` gives an overview of the package.

**Examples**

```
## Not run:
dir <- system.file("example", package="scapeMCMC")
xmcmc <- importMCMC(dir) # or rename and select particular elements:
xmcmc <- importMCMC(dir, pretty.labels=TRUE,
                    l.choose=c("CAC", "CAS", "Survey", "Prior", "Total"),
                    p.choose=c("R0", "Rinit", "uinit", "cSleft", "cSfull",
                               "sSleft", "sSfull", "logq"))

## End(Not run)
```

---

importProj

---

*Import Coleraine MCMC Projections*


---

**Description**

Import Coleraine MCMC traces for spawning biomass and catch, projected into the near future.

**Usage**

```
importProj(dir, info="", coda=FALSE, quiet=TRUE)
```

**Arguments**

<code>dir</code>	directory containing the files 'strategy.out', 'projspb.m.pst' and 'proccatch.pst'.
<code>info</code>	optional string containing information to store with MCMC projections.
<code>coda</code>	whether data frames should be coerced to class <code>mcmc</code> using the <b>coda</b> package.
<code>quiet</code>	whether to report progress while parsing files in directory.

**Value**

A list containing:

B biomass by catch policy and year  
Y catch by catch policy and year

as lists of data frames, or mcmc objects if coda=TRUE.

**Note**

MCMC projections can be used to evaluate the short-term outcome of harvest policies (constant catch or constant harvest rate), given the uncertainty about parameter values and random future recruitment.

The example dataset xproj was generated using importProj.

The functions `ll` (package **gdata**) and `head` are recommended for browsing MCMC projections, e.g. `ll(xproj)`; `ll(xproj$B)`; `ll(xproj$B$"0.2")`; `head(xproj$B$"0.2")`.

**References**

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model*. User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at <http://fish.washington.edu/research/coleraine/coleraine.pdf>.

**See Also**

[read.table](#), [readLines](#), and [scan](#) import any data.

[importMCMC](#) and `importProj` import Coleraine MCMC results.

`xproj` was created using `importProj`.

[scapeMCMC-package](#) gives an overview of the package.

**Examples**

```
## Not run:  
dir <- system.file("example", package="scapeMCMC")  
xproj <- importProj(dir)  
  
## End(Not run)
```

---

`plotAuto`*Plot MCMC Autocorrelation*

---

**Description**

Plot Markov chain Monte Carlo autocorrelation over a range of lag values. This is a diagnostic plot for deciding whether a chain needs further thinning.

**Usage**

```
plotAuto(mcmc, thin=1, log=FALSE, base=10, main=NULL, xlab="Lag",
         ylab="Autocorrelation", lty=1, lwd=1, col="black", ...)
```

**Arguments**

<code>mcmc</code>	MCMC chain(s) as a vector, data frame or <code>mcmc</code> object.
<code>thin</code>	interval to subsample chain(s), or 1 to keep chain intact.
<code>log</code>	whether values should be log-transformed.
<code>base</code>	logarithm base.
<code>main</code>	main title.
<code>xlab</code>	x-axis label.
<code>ylab</code>	y-axis label.
<code>lty</code>	line type.
<code>lwd</code>	line width.
<code>col</code>	line color.
<code>...</code>	passed to <code>autocorr.plot()</code> , <code>title()</code> and <code>axis()</code> .

**Value**

Null, but a plot is drawn on the current graphics device.

**Note**

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

**See Also**

[autocorr.plot](#) is the underlying plotting function, and [window.mcmc](#) is used to optionally thin the chain(s).

[plotTrace](#), [plotAuto](#), [plotCumu](#), and [plotSplom](#) are diagnostic plots.

[plotDens](#) and [plotQuant](#) are posterior plots.

[scapeMCMC-package](#) gives an overview of the package.

**Examples**

```
plotAuto(xmcmc$P$R0)
plotAuto(xmcmc$P$R0, thin=10)
plotAuto(xmcmc$P, lag.max=50, ann=FALSE, axes=FALSE)
```

plotCumulative

*Plot MCMC Cumulative Quantiles***Description**

Plot Markov chain Monte Carlo cumulative quantiles. This is a diagnostic plot for deciding whether the chain has converged.

**Usage**

```
plotCumulative(mcmc, probs=c(0.025,0.975), div=1, log=FALSE, base=10,
               main=NULL, xlab="Iterations", ylab="Value", lty.median=1,
               lwd.median=2, col.median="black", lty.outer=2, lwd.outer=1,
               col.outer="black", ...)
```

**Arguments**

mcmc	MCMC chain(s) as a vector, data frame or mcmc object.
probs	outer quantiles to draw, a vector of length 2.
div	denominator to shorten values on the y axis.
log	whether values should be log-transformed.
base	logarithm base.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
lty.median	line type of median.
lwd.median	line width of median.
col.median	color of median.
lty.outer	line type of outer quantiles.
lwd.outer	line width of outer quantiles.
col.outer	color of outer quantiles.
...	passed to cumuplot(), title() and axis().

**Value**

Null, but a plot is drawn on the current graphics device.

**Note**

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

**See Also**

`cumuplot` is the underlying plotting function, and `quantile` is called iteratively to calculate the cumulative quantiles.

`plotTrace`, `plotAuto`, `plotCumu`, and `plotSplom` are diagnostic plots.

`plotDens` and `plotQuant` are posterior plots.

`scapeMCMC-package` gives an overview of the package.

**Examples**

```
plotCumu(xmcmc$P$R0, main="R0")
plotCumu(xmcmc$P$cSfull, main="cSfull")
plotCumu(xmcmc$P, probs=c(0.25,0.50,0.75), ann=FALSE, axes=FALSE)
```

---

plotDens

*Plot MCMC Density*

---

**Description**

Plot Markov chain Monte Carlo density. This is an approximation of the posterior probability density function.

**Usage**

```
plotDens(mcmc, probs=c(0.025,0.975), points=FALSE, axes=TRUE,
  same.limits=FALSE, between=list(x=axes,y=axes), div=1,
  log=FALSE, base=10, main=NULL, xlab=NULL, ylab=NULL,
  cex.main=1.2, cex.lab=1, cex.axis=0.8, cex.strip=0.8,
  col.strip="gray95", las=0, tck=0.5, tick.number=5,
  lty.density=1, lwd.density=3, col.density="black",
  lty.median=2, lwd.median=1, col.median="darkgray", lty.outer=3,
  lwd.outer=1, col.outer="darkgray", pch="|", cex.points=1,
  col.points="black", plot=TRUE, ...)
```

**Arguments**

<code>mcmc</code>	MCMC chain(s) as a vector, data frame or <code>mcmc</code> object.
<code>probs</code>	vector of outer quantiles to draw, besides the median.
<code>points</code>	whether data points should be plotted along the x axis.
<code>axes</code>	whether axis values should be plotted.
<code>same.limits</code>	whether panels should have same x-axis limits.

between	list with x and y indicating panel spacing.
div	denominator to shorten values on the x axis.
log	whether values should be log-transformed.
base	logarithm base.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.density	line type of density curve.
lwd.density	line width of density curve.
col.density	color of density curve.
lty.median	line type of median.
lwd.median	line width of median.
col.median	color of median.
lty.outer	line type of outer quantiles.
lwd.outer	line width of outer quantiles.
col.outer	color of outer quantiles.
pch	symbol for data points.
cex.points	size of data points.
col.points	color of data points.
plot	whether to draw plot.
...	passed to densityplot and panel.densityplot.

**Value**

When `plot=TRUE`, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When `plot=FALSE`, a trellis object is returned.

**Note**

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

**See Also**

`xyplot` and `panel.densityplot` are the underlying drawing functions, and `link[coda]{densplot}` is a similar non-trellis plot.

`plotTrace`, `plotAuto`, `plotCumu`, and `plotSplom` are diagnostic plots.

`plotDens` and `plotQuant` are posterior plots.

`scapeMCMC-package` gives an overview of the package.

**Examples**

```
plotDens(xmcmc$B$"2004", points=TRUE, div=1000, main="2004\n",
         xlab="Biomass age 4+ (1000 t)", tick.number=6, strip=FALSE)
plotDens(xmcmc$P, xlab="Parameter value", ylab="Posterior density\n")
```

---

plotQuant

*Plot MCMC Quantiles*

---

**Description**

Plot quantiles of multiple Markov chain Monte Carlo chains, using bars, boxes, or lines.

**Usage**

```
plotQuant(mcmc, style="boxes", probs=c(0.025,0.975), axes=TRUE,
          names=NULL, ylim=NULL, yaxs="i", div=1, log=FALSE, base=10,
          main=NULL, xlab=NULL, ylab=NULL, cex.axis=0.8, las=1,
          tck=-0.015, tick.number=8, lty.median=1,
          lwd.median=1+2*(style!="boxes"), col.median="black",
          lty.outer=1+2*(style=="lines"), lwd.outer=1,
          col.outer="darkgray", boxfill="darkgray", boxwex=0.7,
          mai=c(0.8,1,1,0.6),
          mgp=list(bottom=c(2,0.4,0),left=c(3,0.6,0),top=c(0,0.6,0),
                  right=c(0,0.6,0)), ...)
```

**Arguments**

<code>mcmc</code>	MCMC chains as a data frame or <code>mcmc</code> object.
<code>style</code>	how quantiles should be drawn: "bars", "boxes", or "lines".
<code>probs</code>	outer quantiles to draw, a vector of length 2.
<code>axes</code>	numeric vector indicating which axis labels should be drawn: 1=bottom, 2=left, 3=top, 4=right, or TRUE to display all (default).
<code>names</code>	x-axis labels.
<code>ylim</code>	y-axis limits.
<code>yaxs</code>	y-axis style: "i" to truncate exactly at limits (default) or "r" to extend the axis slightly beyond the limits.

div	denominator to shorten values on the y axis.
log	whether values should be log-transformed.
base	logarithm base.
main	main title.
xlab	x-axis label.
ylab	y-axis label.
cex.axis	size of tick labels.
las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.median	line type of median.
lwd.median	line width of median.
col.median	color of median.
lty.outer	line type of outer quantiles.
lwd.outer	line width of outer quantiles.
col.outer	color of outer quantiles.
boxfill	color of boxes.
boxwex	relative width of boxes.
mai	margins around plot as a vector of four numbers (bottom, left, top, right).
mgp	margins around axis titles, labels, and lines as a list of four vectors (bottom, left, top, right).
...	passed to plot, bxp, plotCI, lines, matplot, axis, and title.

**Value**

List containing:

x	midpoint coordinates on the x axis.
y	quantile coordinates on the y axis.

**Note**

The Args function from the **gdata** package is recommended for reviewing the arguments, instead of args.

**See Also**

[bxp](#), [plotCI](#), and [matplot](#) are the underlying drawing functions.  
[plotTrace](#), [plotAuto](#), [plotCumu](#), and [plotSplom](#) are diagnostic plots.  
[plotDens](#) and [plotQuant](#) are posterior plots.  
[scapeMCMC-package](#) gives an overview of the package.

**Examples**

```
plotQuant(xmcmc$B, style="lines", div=1000, xlab="Year",
          ylab="Biomass age 4+ (1000 t)")
plotQuant(xmcmc$R, names=substring(names(xmcmc$R),3), div=1000,
          xlab="Year", ylab="Recruitment (million one-year-olds)")
```

---

plotSplom

*Plot MCMC Scatterplot Matrix*


---

**Description**

Plot scatterplots of multiple Markov chain Monte Carlo chains. This is a diagnostic plot for deciding whether parameters are confounded. When parameter estimates are highly dependent on each other, it may undermine conclusions based on MCMC results of that model.

**Usage**

```
plotSplom(mcmc, axes=FALSE, between=0, div=1, log=FALSE, base=10, ...)
```

**Arguments**

mcmc	MCMC chains as a data frame or mcmc object.
axes	whether axis values should be plotted.
between	space between panels.
div	denominator to shorten values on the y axis.
log	whether values should be log-transformed.
base	logarithm base.
...	passed to pairs().

**Value**

Null, but a plot is drawn on the current graphics device.

**Note**

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

**See Also**

[pairs](#) is the underlying drawing function, and [splom](#) is a similar trellis plot. [plotTrace](#), [plotAuto](#), [plotCumu](#), and [plotSplom](#) are diagnostic plots. [plotDens](#) and [plotQuant](#) are posterior plots. [scapeMCMC-package](#) gives an overview of the package.

**Examples**

```
plotSplom(xmcmc$P)
plotSplom(xproj$B$"0.25", axes=TRUE, between=1, div=1000,
          main="Future biomass", cex.labels=1.5)
```

plotTrace

*Plot MCMC Traces***Description**

Plot Markov chain Monte Carlo traces. This is a diagnostic plot for deciding whether a chain shows unwanted trends.

**Usage**

```
plotTrace(mcmc, axes=FALSE, same.limits=FALSE,
          between=list(x=axes,y=axes), div=1, span=1/4, log=FALSE,
          base=10, main=NULL, xlab=NULL, ylab=NULL, cex.main=1.2,
          cex.lab=1, cex.axis=0.8, cex.strip=0.8, col.strip="gray95",
          las=0, tck=0.5, tick.number=5, lty.trace=1, lwd.trace=1,
          col.trace="gray", lty.median=1, lwd.median=1,
          col.median="black", lty.loess=2, lwd.loess=1,
          col.loess="black", plot=TRUE, ...)
```

**Arguments**

mcmc	MCMC chain(s) as a vector, data frame or mcmc object.
axes	whether axis values should be plotted.
same.limits	whether panels should have same x-axis limits.
between	list with x and y indicating panel spacing.
div	denominator to shorten values on the y axis.
span	smoothness parameter, passed to panel.loess
log	whether values should be log-transformed.
base	logarithm base.
main	main title.
xlab	x-axis title.
ylab	y-axis title.
cex.main	size of main title.
cex.lab	size of axis labels.
cex.axis	size of tick labels.
cex.strip	size of strip labels.
col.strip	color of strip labels.

las	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
tck	tick mark length.
tick.number	number of tick marks.
lty.trace	line type of trace.
lwd.trace	line width of trace.
col.trace	color of trace.
lty.median	line type of median.
lwd.median	line width of median.
col.median	color of median.
lty.loess	line type of loess.
lwd.loess	line width of loess.
col.loess	color of loess.
plot	whether to draw plot.
...	passed to <code>xyplot</code> and <code>panel.loess</code> .

### Value

When `plot=TRUE`, a trellis plot is drawn and a data frame is returned, containing the data used for plotting. When `plot=FALSE`, a trellis object is returned.

### Note

The `Args` function from the **gdata** package is recommended for reviewing the arguments, instead of `args`.

### See Also

[xyplot](#) and [panel.loess](#) are the underlying drawing functions, and [traceplot](#) is a similar non-trellis plot.

`plotTrace`, [plotAuto](#), [plotCumu](#), and [plotSplom](#) are diagnostic plots.

[plotDens](#) and [plotQuant](#) are posterior plots.

[scapeMCMC-package](#) gives an overview of the package.

### Examples

```
plotTrace(xmcmc$P, xlab="Iterations", ylab="Parameter value",
          layout=c(2,4))
plotTrace(xmcmc$P$R0, axes=TRUE, div=1000)
```

---

`xmcmc`*MCMC Results from Cod Assessment*

---

## Description

Markov chain Monte Carlo results from stock assessment of cod (*Gadus morhua*) in Icelandic waters.

## Usage

```
xmcmc
```

## Format

List containing four data frames:

- L likelihood components: CAC (commercial catch at age), CAs (survey catch at age), Survey (survey abundance index), Prior
- P estimated parameters: R0 (average virgin recruitment), Rinit (initial recruitment scaler), uinit (initial harvest rate), cSle
- B predicted biomass (age 4+) by year.
- R predicted recruitment by year.

## Details

Some aspects of the model are described on the [x.cod](#) help page in the **scape** package.

## Note

The list was imported from the files 'mcmclike.out', 'params.pst', 'spawbiom.pst' and 'recruits.pst', using the `importMCMC()` function. These files can be found in the 'scapeMCMC/example' directory.

The functions `ll` (package **gdata**) and `head` are recommended for browsing MCMC results, e.g. `ll(xmcmc)`; `ll(xmcmc$P)`; `head(xmcmc$P)`.

## References

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model*. User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at <http://fish.washington.edu/research/coleraine/coleraine.pdf>.

Magnusson, A. 2003. *Coleraine assessment of the Icelandic cod stock*. Report for the Icelandic Marine Research Institute. Available from the author.

## See Also

[importMCMC](#) was used to import the MCMC results.

`xmcmc` and `xproj` are MCMC results and projections to explore.

[scapeMCMC-package](#) gives an overview of the package.

**Examples**

```

plotTrace(xmcmc$L)
plotAuto(xmcmc$P)
plotCumu(xmcmc$P)
plotSplom(xmcmc$P)
plotDens(xmcmc$P)
plotQuant(xmcmc$B)

```

---

xproj

MCMC Projections from Cod Assessment

---

**Description**

Markov chain Monte Carlo projections from stock assessment of cod (*Gadus morhua*) in Icelandic waters.

**Usage**

```
xproj
```

**Format**

List containing two lists:

- B projected biomass by year, given a constant harvest rate policy: "0", "0.05", ..., "0.50".
- Y projected catch by year, given a constant harvest rate policy: "0", "0.05", ..., "0.50".

**Note**

MCMC projections can be used to evaluate the short-term outcome of harvest policies (constant catch or constant harvest rate), given the uncertainty about parameter values and random future recruitment. Some aspects of the model are described on the [xmcmc](#) help page, and on the [x.cod](#) help page in the **scape** package.

The list was imported from the files 'strategy.out', 'projspbm.out' and 'procatch.out', using the `importProj()` function. These files can be found in the 'scapeMCMC/example' directory.

The functions `ll` (package **gdata**) and `head` are recommended for browsing MCMC projections, e.g. `ll(xproj)`; `ll(xproj$B)`; `ll(xproj$B$"0.2")`; `head(xproj$B$"0.2")`.

**References**

Hilborn, R., M. Maunder, A. Parma, B. Ernst, J. Payne, and P. Starr. 2003. *Coleraine: A generalized age-structured stock assessment model*. User's manual version 2.0. University of Washington Report SAFS-UW-0116. Available at <http://fish.washington.edu/research/coleraine/coleraine.pdf>.

Magnusson, A. 2003. *Coleraine assessment of the Icelandic cod stock*. Report for the Icelandic Marine Research Institute. Available from the author.

**See Also**

[importProj](#) was used to import the MCMC projections.

`xmcmc` and `xproj` are MCMC results and projections to explore.

[scapeMCMC-package](#) gives an overview of the package.

**Examples**

```
plotTrace(xproj$B$"0.25")  
plotAuto(xproj$B$"0.25")  
plotCumu(xproj$B$"0.25")  
plotSpIom(xproj$B$"0.25")  
plotDens(xproj$B$"0.25")  
plotQuant(xproj$B$"0.25")
```

# Index

- \*Topic **datasets**
  - xmcmc, 15
  - xproj, 16
- \*Topic **file**
  - importMCMC, 3
  - importProj, 4
- \*Topic **hplot**
  - plotAuto, 6
  - plotCumulative, 7
  - plotDens, 8
  - plotQuant, 10
  - plotSplom, 12
  - plotTrace, 13
  - scapeMCMC-package, 2
- \*Topic **interface**
  - importMCMC, 3
  - importProj, 4
- \*Topic **ts**
  - plotAuto, 6
  - plotCumulative, 7
  - plotDens, 8
  - plotQuant, 10
  - plotSplom, 12
  - plotTrace, 13
  - scapeMCMC-package, 2
- autocorr.plot, 6
- bxp, 11
- cumuplot, 8
- importMCMC, 2, 3, 5, 15
- importProj, 2, 4, 4, 17
- matplot, 11
- pairs, 12
- panel.densityplot, 10
- panel.loess, 14
- plotAuto, 2, 6, 8, 10–12, 14
- plotCI, 11
- plotCumulative, 2, 6, 7, 10–12, 14
- plotDens, 2, 6, 8, 8, 11, 12, 14
- plotQuant, 2, 6, 8, 10, 10, 12, 14
- plotSplom, 2, 6, 8, 10, 11, 12, 14
- plotTrace, 2, 6, 8, 10–12, 13
- quantile, 8
- read.table, 4, 5
- readLines, 4, 5
- scan, 4, 5
- scapeMCMC (scapeMCMC-package), 2
- scapeMCMC-package, 4–6, 8, 10–12, 14, 15, 17
- scapeMCMC-package, 2
- splom, 12
- traceplot, 14
- window.mcmc, 6
- x.cod, 15, 16
- xmcmc, 2, 4, 15, 16
- xproj, 2, 5, 15, 16, 17
- xyplot, 10, 14