

# Package: plotMCMC (via r-universe)

September 16, 2024

**Version** 2.0.1

**Date** 2020-11-23

**Title** MCMC Diagnostic Plots

**Imports** coda, gplots, lattice

**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.

**License** GPL-3

**URL** <https://github.com/arni-magnusson/plotMCMC>

**Repository** <https://arni-magnusson.r-universe.dev>

**RemoteUrl** <https://github.com/arni-magnusson/plotmcmc>

**RemoteRef** HEAD

**RemoteSha** 4a3a7367c97de878147884696082c970f4525b02

## Contents

plotMCMC-package	2
plotAuto	3
plotCumulative	4
plotDens	5
plotQuant	7
plotSplom	9
plotTrace	10
xbio	12
xpar	13
xpro	15
xrec	16
<b>Index</b>	<b>18</b>

---

plotMCMC-package      *MCMC Diagnostic Plots*

---

## 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.

## Details

*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>xpar</code>	model parameters
<code>xrec</code>	recruitment
<code>xbio</code>	biomass
<code>xpro</code>	future projected biomass

## Note

`browseVignettes()` shows a vignette with all the example plots.

The plot functions assume that MCMC results are stored either as a plain `numeric` vector (single chain) or in a `data.frame` (multiple chains). The `mcmc` class is also supported.

## Author(s)

Arni Magnusson and Ian Stewart.

## References

Fournier, D.A., Skaug, H.J., Ancheta, J., Iannelli, J., Magnusson, A., Maunder, M.N., Nielsen, A., and Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, **27**, 233–249. doi:10.1080/10556788.2011.597854

Magnusson, A., Punt, A.E., and Hilborn, R. (2013). Measuring uncertainty in fisheries stock assessment: the delta method, bootstrap, and MCMC. *Fish and Fisheries*, **14**, 325–342. doi:10.1111/j.14672979.2012.00473.x

**See Also**

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

Many **plotMCMC** graphics are trellis plots, rendered with the **lattice** package.

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

---

 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.

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

**Examples**

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

---

plotCumu

*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**

```
plotCumu(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**

<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>div</code>	denominator to shorten values on the y axis.
<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.median</code>	line type of median.
<code>lwd.median</code>	line width of median.
<code>col.median</code>	color of median.
<code>lty.outer</code>	line type of outer quantiles.
<code>lwd.outer</code>	line width of outer quantiles.
<code>col.outer</code>	color of outer quantiles.
<code>...</code>	passed to <code>cumuplot</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**

[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.

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

**Examples**

```
plotCumu(xpar$R0, main="R0")
plotCumu(xpar$cSfull, main="cSfull")
plotCumu(xpar, probs=c(0.25,0.75), 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 individual 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.
<code>between</code>	list with x and y indicating panel spacing.
<code>div</code>	denominator to shorten values on the x axis.
<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>cex.main</code>	size of main title.
<code>cex.lab</code>	size of axis labels.
<code>cex.axis</code>	size of tick labels.
<code>cex.strip</code>	size of strip labels.
<code>col.strip</code>	color of strip labels.
<code>las</code>	orientation of tick labels: 0=parallel, 1=horizontal, 2=perpendicular, 3=vertical.
<code>tck</code>	tick mark length.
<code>tick.number</code>	number of tick marks.
<code>lty.density</code>	line type of density curve.
<code>lwd.density</code>	line width of density curve.
<code>col.density</code>	color of density curve.
<code>lty.median</code>	line type of median.
<code>lwd.median</code>	line width of median.
<code>col.median</code>	color of median.
<code>lty.outer</code>	line type of outer quantiles.
<code>lwd.outer</code>	line width of outer quantiles.
<code>col.outer</code>	color of outer quantiles.
<code>pch</code>	symbol for points.
<code>cex.points</code>	size of points.
<code>col.points</code>	color of points.
<code>plot</code>	whether to draw plot.
<code>...</code>	passed to <code>densityplot</code> and <code>panel.densityplot</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.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.

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

**Examples**

```
plotDens(xbio$"2004", points=TRUE, div=1000, main="2004\n",
         xlab="Biomass age 4+ (kt)", tick.number=6, strip=FALSE)
plotDens(xpar, 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*(style!="bars"),
         lwd.median=1+1*(style!="boxes"), col.median="black",
         lty.outer=1+2*(style=="lines"), lwd.outer=1,
         col.outer="black", pch=16, cex=0.8, col="black",
         boxfill="darkgray", boxwex=0.5, staplewex=0.5, sfrac=0.005,
         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**

mcmc	MCMC chains as a data frame or mcmc object.
style	how quantiles should be drawn: "bars", "boxes", or "lines".
probs	vector of outer quantiles to draw, besides the median.
axes	numeric vector indicating which axis labels should be drawn: 1=bottom, 2=left, 3=top, 4=right, or TRUE to display all (default).
names	x-axis labels.
ylim	y-axis limits.
yaxs	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.
pch	symbol for points.
cex	size of points.
col	color of points.
boxfill	color of boxes.
boxwex	width of boxes.
staplewex	width of error bar staples when style="boxes", as a fraction of box width.
sfrac	width of error bar staples when style="bars", as a fraction of plot region.
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**

With `style="boxes"`, the quartiles are shown as boxes.

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.

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

**Examples**

```
plotQuant(xrec, names=substring(names(xrec),3), div=1000, xlab="Year",
          ylab="Recruitment (million one-year-olds)")
plotQuant(xbio, div=1000, xlab="Year", ylab="Biomass age 4+ (kt)")
plotQuant(xbio, style="bars", div=1000, sfrac=0, xlab="Year",
          ylab="Biomass age 4+ (kt)")
plotQuant(xbio, style="lines", div=1000, xlab="Year",
          ylab="Biomass age 4+ (kt)")
plotQuant(xpro, axes=1:2, div=1000, xlab="Year",
          ylab="Biomass age 4+ (kt)")
```

---

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.

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

**Examples**

```
plotSplom(xpar, pch=".")
plotSplom(xpro, axes=TRUE, between=1, div=1000, main="Future biomass",
          cex.labels=1.5, pch=".", cex=3)
```

---

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 xypplot and panel.loess.

**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.

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

**Examples**

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

---

xbio

*MCMC Results for Biomass*

---

**Description**

Markov chain Monte Carlo results from stock assessment of cod (*Gadus morhua*) in Icelandic waters, showing estimated biomass by year in tonnes.

**Usage**

```
xbio
```

**Format**

Data frame containing 1000 rows and 34 columns (years 1971 to 2004).

**Details**

Each column contains the results of 1 million MCMC iterations, after thinning to every 1000th iteration.

The MCMC analysis started at the best fit, so no burn-in period was discarded.

**Note**

Biomass is the total weight of all individuals in a population, in this case ages 4 and older.

This data frame is a subset of the `xmcmc` list from the **scape** package, which contains further documentation about the data and model. Specifically, `xbio <- xmcmc$B`.

The MCMC analysis was run using the *AD Model Builder* software (<https://www.admb-project.org>).

**References**

Fournier, D.A., Skaug, H.J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M.N., Nielsen, A., and Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, **27**, 233–249. doi:10.1080/10556788.2011.597854

Magnusson, A., Punt, A.E., and Hilborn, R. (2013). Measuring uncertainty in fisheries stock assessment: the delta method, bootstrap, and MCMC. *Fish and Fisheries*, **14**, 325–342. doi:10.1111/j.14672979.2012.00473.x

**See Also**

`xpar` (parameters), `xrec` (recruitment), `xbio` (biomass), and `xpro` (projected future biomass) are MCMC data frames to explore.

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

**Examples**

```
plotDens(xbio$"2004", points=TRUE, div=1000, main="2004\n",
         xlab="Biomass age 4+ (1000 t)", tick.number=6, strip=FALSE)

plotQuant(xbio, div=1000, xlab="Year", ylab="Biomass age 4+ (kt)")
plotQuant(xbio, style="bars", div=1000, sfrac=0, xlab="Year",
         ylab="Biomass age 4+ (kt)")
plotQuant(xbio, style="lines", div=1000, xlab="Year",
         ylab="Biomass age 4+ (kt)")
```

---

xpar

*MCMC Results for Model Parameters*


---

**Description**

Markov chain Monte Carlo results from stock assessment of cod (*Gadus morhua*) in Icelandic waters, showing estimated model parameters.

**Usage**

```
xpar
```

## Format

Data frame containing 1000 rows and 8 columns:

R0	average virgin recruitment
Rinit	initial recruitment scaler
uinit	initial harvest rate
cSleft	left-side slope of commercial selectivity curve
cSfull	age at full commercial selectivity
sSleft	left-side slope of survey selectivity curve
sSfull	age at full survey selectivity
logq	log-transformed survey catchability

## Details

Each column contains the results of 1 million MCMC iterations, after thinning to every 1000th iteration.

The MCMC analysis started at the best fit, so no burn-in period was discarded.

## Note

This data frame is a subset of the `xmcmc` list from the **scape** package, which contains further documentation about the data and model. Specifically, `xpar <- xmcmc$P`.

The MCMC analysis was run using the *AD Model Builder* software (<https://www.admb-project.org>).

## References

Fournier, D.A., Skaug, H.J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M.N., Nielsen, A., and Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, **27**, 233–249. doi:10.1080/10556788.2011.597854

Magnusson, A., Punt, A.E., and Hilborn, R. (2013). Measuring uncertainty in fisheries stock assessment: the delta method, bootstrap, and MCMC. *Fish and Fisheries*, **14**, 325–342. doi:10.1111/j.14672979.2012.00473.x

## See Also

`xpar` (parameters), `xrec` (recruitment), `xbio` (biomass), and `xpro` (projected future biomass) are MCMC data frames to explore.

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

## Examples

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

plotAuto(xpar$R0)
```

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

plotCumu(xpar$R0, main="R0")
plotCumu(xpar$cSfull, main="cSfull")
plotCumu(xpar, probs=c(0.25,0.75), ann=FALSE, axes=FALSE)

plotSplom(xpar, pch=".")

plotDens(xpar, xlab="Parameter value", ylab="Posterior density\n")
```

---

xpro

*MCMC Results for Future Projections*

---

### Description

Markov chain Monte Carlo results from stock assessment of cod (*Gadus morhua*) in Icelandic waters, showing future projected biomass in tonnes.

### Usage

xpro

### Format

Data frame containing 1000 rows and 4 columns (years 2004 to 2007).

### Details

Each column contains the results of 1 million MCMC iterations, after thinning to every 1000th iteration.

The MCMC analysis started at the best fit, so no burn-in period was discarded.

### Note

The projections are based on a fixed harvest rate, where 25% of the biomass (ages 4 and older) is caught every year.

This data frame is a subset of the xproj list from the **scape** package, which contains further documentation about the data and model. Specifically, `xpro <- xproj$"0.25"`.

The MCMC analysis was run using the *AD Model Builder* software (<https://www.admb-project.org>).

## References

Fournier, D.A., Skaug, H.J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M.N., Nielsen, A., and Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, **27**, 233–249. doi:10.1080/10556788.2011.597854

Magnusson, A., Punt, A.E., and Hilborn, R. (2013). Measuring uncertainty in fisheries stock assessment: the delta method, bootstrap, and MCMC. *Fish and Fisheries*, **14**, 325–342. doi:10.1111/j.14672979.2012.00473.x

## See Also

`xpar` (parameters), `xrec` (recruitment), `xbio` (biomass), and `xpro` (projected future biomass) are MCMC data frames to explore.

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

## Examples

```
plotQuant(xpro, axes=1:2, div=1000, xlab="Year",
          ylab="Biomass age 4+ (kt)")
```

```
plotSplom(xpro, axes=TRUE, between=1, div=1000, main="Future biomass",
          cex.labels=1.5, pch=".", cex=3)
```

---

xrec

*MCMC Results for Recruitment*

---

## Description

Markov chain Monte Carlo results from stock assessment of cod (*Gadus morhua*) in Icelandic waters, showing estimated recruitment by year.

## Usage

```
xrec
```

## Format

Data frame containing 1000 rows and 33 columns (years 1970 to 2002).

## Details

Each column contains the results of 1 million MCMC iterations, after thinning to every 1000th iteration.

The MCMC analysis started at the best fit, so no burn-in period was discarded.

## Note

Recruitment is the size of a cohort (year class), in this case thousands of one-year-olds.

For example, `xrec$"1980"` is the estimated number of one-year-olds in 1981, the cohort that hatched in 1980.

This data frame is a subset of the `xmcmc` list from the **scape** package, which contains further documentation about the data and model. Specifically, `xrec <- xmcmc$R`.

The MCMC analysis was run using the *AD Model Builder* software (<https://www.admb-project.org>).

## References

Fournier, D.A., Skaug, H.J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M.N., Nielsen, A., and Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, **27**, 233–249. doi:10.1080/10556788.2011.597854

Magnusson, A., Punt, A.E., and Hilborn, R. (2013). Measuring uncertainty in fisheries stock assessment: the delta method, bootstrap, and MCMC. *Fish and Fisheries*, **14**, 325–342. doi:10.1111/j.14672979.2012.00473.x

## See Also

`xpar` (parameters), `xrec` (recruitment), `xbio` (biomass), and `xpro` (projected future biomass) are MCMC data frames to explore.

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

## Examples

```
plotQuant(xrec, names=substring(names(xrec),3), div=1000, xlab="Year",
          ylab="Recruitment (million one-year-olds)")
```

# Index

autocorr.plot, 4

bxp, 9

cumuplot, 5

data.frame, 2

matplot, 9

mcmc, 2

numeric, 2

pairs, 10

panel.densityplot, 7

panel.loess, 12

plotAuto, 2, 3, 5, 7, 9, 10, 12

plotCI, 9

plotCumu, 2, 4, 4, 7, 9, 10, 12

plotDens, 2, 4, 5, 5, 9, 10, 12

plotMCMC (plotMCMC-package), 2

plotMCMC-package, 2

plotQuant, 2, 4, 5, 7, 7, 10, 12

plotSplom, 2, 4, 5, 7, 9, 9, 12

plotTrace, 2, 4, 5, 7, 9, 10, 10

quantile, 5

splom, 10

traceplot, 12

window.mcmc, 4

xbio, 2, 12, 14, 17

xpar, 2, 13, 13, 16, 17

xpro, 2, 13, 14, 15, 16, 17

xrec, 2, 13, 14, 16, 16

xyplot, 7, 12