

Package ‘stochvol’

February 18, 2016

Encoding UTF-8

Type Package

Title Efficient Bayesian Inference for Stochastic Volatility (SV) Models

Version 1.2.3

Maintainer Gregor Kastner <gregor.kastner@wu.ac.at>

Description Efficient algorithms for fully Bayesian estimation of stochastic volatility (SV) models via Markov chain Monte Carlo (MCMC) methods.

License GPL (>= 2)

Depends R (>= 3.0.2), coda

Imports Rcpp (>= 0.11), methods, stats, graphics, utils

Suggests mvtnorm

LinkingTo Rcpp, RcppArmadillo (>= 0.4)

NeedsCompilation yes

Author Gregor Kastner [aut, cre]

Repository CRAN

Date/Publication 2016-02-18 08:56:13

R topics documented:

stochvol-package	2
arpredict	3
exrates	4
Extractors	5
logret	6
paradensplot	6
paratraceplot	8
plot.svdraws	9
predict.svdraws	11
svsample	12
svsample2	16

svsim	18
updatesummary	19
volplot	21

Index	24
--------------	-----------

stochvol-package *Efficient Bayesian Inference for Stochastic Volatility (SV) Models*

Description

This package provides an efficient algorithm for fully Bayesian estimation of stochastic volatility (SV) models via Markov chain Monte Carlo (MCMC) methods. Algorithmic details can be found in Kastner and Frühwirth-Schnatter (2014).

Details

Bayesian inference for stochastic volatility models using MCMC methods highly depends on actual parameter values in terms of sampling efficiency. While draws from the posterior utilizing the standard centered parameterization break down when the volatility of volatility parameter in the latent state equation is small, non-centered versions of the model show deficiencies for highly persistent latent variable series. The novel approach of ancillarity-sufficiency interweaving (Yu and Meng, 2011) has recently been shown to aid in overcoming these issues for a broad class of multilevel models. This package provides software for “combining best of different worlds” which allows for inference for parameter constellations that have previously been infeasible to estimate without the need to select a particular parameterization beforehand.

Note

This package is currently in active development. Your comments, suggestions and requests are warmly welcome!

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

References

- Kastner, G. and Frühwirth-Schnatter, S. (2014). Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, **76**, 408–423, <http://dx.doi.org/10.1016/j.csda.2013.01.002>.
- Yu, Y. and Meng, X.-L. (2011). To Center or Not to Center: That is Not the Question—An Ancillarity-Sufficiency Interweaving Strategy (ASIS) for Boosting MCMC Efficiency. *Journal of Computational and Graphical Statistics*, **20**, 531–570, <http://dx.doi.org/10.1198/jcgs.2011.203main>.

Examples

```
## Simulate a highly persistent SV process
sim <- svsim(500, mu = -10, phi = 0.99, sigma = 0.2)

## Obtain 4000 draws from the sampler (that's too little!)
draws <- svsample(sim$y, draws = 4000, burnin = 100, priormu = c(-10, 1),
  priorphi = c(20, 1.2), priorsigma = 0.2)

## Predict 20 days ahead
fore <- predict(draws, 20)

## plot the results
plot(draws, forecast = fore)
```

arpredict

Dynamic prediction for the AR-SV model

Description

Simulates draws from the posterior predictive density of a fitted AR-SV model.

Usage

```
arpredict(object, volpred)
```

Arguments

object	svdraws object as returned from svsample .
volpred	svpredict object as returned from predict.svdraws .

Value

Returns an object of class `c("distpredict", "mcmc")` containing simulations from the posterior predictive density of $y_{(n+1)}, \dots, y_{(n+steps)}$.

Note

You can use the usual coda methods for mcmc objects to print, plot, or summarize the predictions.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[predict.svdraws](#).

Examples

```
## Not run:
data(exrates)
y <- exrates$USD

## Fit AR(1)-SV model to EUR-USD exchange rates
res <- svsample(y, designmatrix = "ar1")

## Use predict.svdraws to obtain predictive volatilities
ahead <- 100
predvol <- predict(res, steps = ahead)

## Use arpredict to obtain draws from the posterior predictive
preddraws <- arpredict(res, predvol)

## Calculate predictive quantiles
predquants <- apply(preddraws, 2, quantile, c(.1, .5, .9))

## Visualize
ts.plot(y, xlim = c(length(y) - ahead, length(y) + ahead),
        ylim = range(predquants))
for (i in 1:3) {
  lines((length(y) + 1):(length(y) + ahead), predquants[i,],
        col = 3, lty = i)
}

## End(Not run)
```

exrates

Euro exchange rate data

Description

The data set contains the daily bilateral prices of one Euro in 23 currencies from January 3, 2000, until April 4, 2012. Conversions to New Turkish Lira and Fourth Romanian Leu have been incorporated.

Usage

```
data(exrates)
```

Source

ECB Statistical Data Warehouse (<http://sdw.ecb.europa.eu>)

See Also

[svsample](#)

Examples

```
## Not run:
data(exrates)
dat <- logret(exrates$USD, demean = TRUE) ## de-meaned log-returns
res <- svsample(dat)                    ## run MCMC sampler
plot(res, forecast = 100)              ## display results

## End(Not run)
```

Extractors

Common Extractors for 'svdraws' Objects

Description

Some simple extractors returning the corresponding element of an svdraws object.

Usage

```
para(x)
latent(x)
latent0(x)
priors(x)
thinning(x)
runtime(x)
```

Arguments

x svdraws object.

Value

The return value depends on the actual function:

para(x)	extracts the parameter draws and returns them as an mcmc object.
latent(x)	extracts the latent contemporaneous log-volatility draws and returns them as an mcmc object.
latent0(x)	extracts the latent initial log-volatility draws and returns as an mcmc object.
priors(x)	extracts the prior parameters used and returns them in a list.
thinning(x)	extracts the thinning parameters used and returns them in a list.
runtime(x)	extracts the runtime and returns it as a proc_time object.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

logret *Computes (de-meaned) log returns.*

Description

Small utility function returning either $\text{diff}(\log(x))$ in case the argument `demean` is set to `FALSE`, or $\text{diff}(\log(x)) - \text{mean}(\text{diff}(\log(x)))$ in case that `demean` is `TRUE`.

Usage

```
logret(x, demean = FALSE)
```

Arguments

<code>x</code>	Real-valued vector.
<code>demean</code>	A single logical value indicating whether the returns should be de-meaned. Defaults to <code>FALSE</code> .

Value

A vector of length $\text{length}(x) - 1$, containing (de-meaned) returns.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

paradensplot *Probability Density Function Plot for the Parameter Posteriors*

Description

Displays a plot of the density estimate for the posterior distribution of the parameters `mu`, `phi`, `sigma` (and potentially `nu`), computed by the [density](#) function.

Usage

```
paradensplot(x, showobs = TRUE, showprior = TRUE, showxlab = TRUE,  
  mar = c(1.9, 1.9, 1.9, 0.5), mgp = c(2, 0.6, 0),  
  simobj = NULL, ...)
```

Arguments

x	svdraws object.
showobs	logical value, indicating whether the observations should be displayed along the x-axis. If many draws have been obtained, the default (TRUE) can render the plotting to be quite slow, and you might want to try setting showobs to FALSE.
showprior	logical value, indicating whether the prior distribution should be displayed. The default value is TRUE.
showxlab	logical value, indicating whether the x-axis should be labelled with the number of iterations and the bandwidth obtained from density . The default value is TRUE.
mar	numerical vector of length 4, indicating the plot margins. See par for details. The default value is <code>c(1.9, 1.9, 1.9, 0.5)</code> , which is slightly smaller than the R-defaults.
mgp	numerical vector of length 3, indicating the axis and label positions. See par for details. The default value is <code>c(2, 0.6, 0)</code> , which is slightly smaller than the R-defaults.
simobj	object of class <code>svsim</code> as returned by the SV simulation function svsim . If provided, “true” data generating values will be added to the plots.
...	further arguments are passed on to the invoked plot function.

Details

`paradensplot` is modeled after [densplot](#) in the coda package, with some modifications for parameters that have (half-)bounded support.

Value

Called for its side effects. Returns argument `x` invisibly.

Note

You can call this function directly, but it is more commonly called by the [plot.svdraws](#) method.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[paratraceplot](#), [volplot](#), [plot.svdraws](#)

paratraceplot

Trace Plot of MCMC Draws from the Parameter Posteriors

Description

Displays a plot of iterations vs. sampled values the parameters μ , ϕ , σ (and potentially ν), with a separate plot per variable.

Usage

```
paratraceplot(x, mar = c(1.9, 1.9, 1.9, 0.5), mgp = c(2, 0.6, 0), simobj = NULL, ...)
```

Arguments

x	svdraws object.
mar	numerical vector of length 4, indicating the plot margins. See par for details. The default value is <code>c(1.9, 1.9, 1.9, 0.5)</code> , which is slightly smaller than the R-defaults.
mgp	numerical vector of length 3, indicating the axis and label positions. See par for details. The default value is <code>c(2, 0.6, 0)</code> , which is slightly smaller than the R-defaults.
simobj	object of class <code>svsim</code> as returned by the SV simulation function svsim . If provided, “true” data generating values will be added to the plots.
...	further arguments are passed on to the invoked <code>matplot</code> function.

Details

`paratraceplot` is modeled after [traceplot](#) in the `coda` package, with very minor modifications.

Value

Called for its side effects. Returns argument `x` invisibly.

Note

You can call this function directly, but it is more commonly called by the [plot.svdraws](#) method.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[paradensplot](#), [volplot](#), [plot.svdraws](#)

Description

plot.svdraws generates some plots visualizing the posterior distribution and can also be used to display predictive distributions of future volatilities.

Usage

```
## S3 method for class 'svdraws'
plot(x, forecast = NULL, dates = NULL, show0 = FALSE, showobs = TRUE,
      showprior = TRUE, col = NULL, forecastlty = NULL, tcl = -0.4,
      mar = c(1.9, 1.9, 1.7, 0.5), mgp = c(2, 0.6, 0), simobj = NULL, ...)
```

Arguments

x	svdraws object.
forecast	nonnegative integer or object of class svpredict, as returned by predict.svdraws . If an integer greater than 0 is provided, predict.svdraws is invoked to obtain the forecast-step-ahead prediction. The default value is 0.
dates	vector of length ncol(x\$latent), providing optional dates for labelling the x-axis. The default value is NULL; in this case, the axis will be labelled with numbers.
show0	logical value, indicating whether the initial volatility $\exp(h_0/2)$ should be displayed. The default value is FALSE.
showobs	logical value, indicating whether the observations should be displayed along the x-axis. If many draws have been obtained, the default (TRUE) can render the plotting to be quite slow, and you might want to try setting showobs to FALSE.
showprior	logical value, indicating whether the prior distribution should be displayed. The default value is TRUE.
col	vector of color values (see par) used for plotting the quantiles. The default value NULL results in gray lines for all quantiles except the median, which is displayed in black.
forecastlty	vector of line type values (see par) used for plotting quantiles of predictive distributions. The default value NULL results in dashed lines.
tcl	The length of tick marks as a fraction of the height of a line of text. See par for details. The default value is -0.4, which results in slightly shorter tick marks than usual.
mar	numerical vector of length 4, indicating the plot margins. See par for details. The default value is c(1.9, 1.9, 1.9, 0.5), which is slightly smaller than the R-defaults.
mgp	numerical vector of length 3, indicating the axis and label positions. See par for details. The default value is c(2, 0.6, 0), which is slightly smaller than the R-defaults.

simobj object of class `svsim` as returned by the SV simulation function `svsim`. If provided, the “true” data generating values will be added to the plots.

... further arguments are passed on to the invoked plotting functions.

Details

This function sets up the page layout and calls `volplot`, `paratraceplot` and `paradensplot`.

Value

Called for its side effects. Returns argument `x` invisibly.

Note

In case you want different quantiles to be plotted, use `updatesummary` on the `svdraws` object first. An example of doing so is given in the Examples section.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[updatesummary](#), [volplot](#), [paratraceplot](#), [paradensplot](#).

Examples

```
## Simulate a short and highly persistent SV process
sim <- svsim(100, mu = -10, phi = 0.99, sigma = 0.2)

## Obtain 5000 draws from the sampler (that's not a lot)
draws <- svsample(sim$y, draws = 5000, burnin = 1000,
  priormu = c(-10, 1), priorphi = c(20, 1.5), priorsigma = 0.2)

## Plot the latent volatilities and some forecasts
plot(draws, forecast = 10)

## Re-plot with different quantiles
newquants <- c(0.01, 0.05, 0.25, 0.5, 0.75, 0.95, 0.99)
draws <- updatesummary(draws, quantiles = newquants)

plot(draws, forecast = 20, showobs = FALSE, col = seq(along = newquants),
  forecastlty = 3, showprior = FALSE)
```

predict.svdraws	<i>Prediction of Future Log-Volatilities</i>
-----------------	--

Description

Simulates draws from the predictive density of the latent log-volatility process.

Usage

```
## S3 method for class 'svdraws'  
predict(object, steps = 1, ...)
```

Arguments

object	svdraws object.
steps	single number, coercible to integer. Denotes the number of steps to forecast.
...	currently ignored.

Value

Returns an object of class `c("svpredict", "mcmc")` containing simulations from the predictive density of $h_{(n+1)}, \dots, h_{(n+steps)}$.

Note

You can use the usual coda methods for mcmc objects to print, plot, or summarize the predictions, or use them within [volplot](#) or [plot.svdraws](#).

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[plot.svdraws](#), [volplot](#).

Examples

```
## Simulate a short and highly persistent SV process  
sim <- svsim(100, mu = -10, phi = 0.99, sigma = 0.2)  
  
## Obtain 5000 draws from the sampler (that's not a lot)  
draws <- svsample(sim$y, draws = 5000, burnin = 100,  
  priormu = c(-10, 1), priorphi = c(20, 1.5), priorsigma = 0.2)  
  
## Predict 10 days ahead  
fore <- predict(draws, 10)
```

```
## Check out the results
summary(fore)
plot(draws, forecast = fore)
```

svsample	<i>Markov Chain Monte Carlo (MCMC) Sampling for the Stochastic Volatility (SV) Model</i>
----------	--

Description

svsample simulates from the joint posterior distribution of the SV parameters μ , ϕ , σ (and potentially ν), along with the latent log-volatilities h_0, \dots, h_n and returns the MCMC draws. If a design matrix is provided, simple Bayesian regression can also be conducted.

Usage

```
svsample(y, draws = 10000, burnin = 1000, designmatrix = NA, priormu = c(0, 100),
  priorphi = c(5, 1.5), priorsigma = 1, priornu = NA, priorbeta = c(0, 10000),
  thinpara = 1, thinlatent = 1, thintime = 1, keptau = FALSE, quiet = FALSE,
  startpara, startlatent, expert, ...)
```

Arguments

y	numeric vector containing the data (usually log-returns), which must not contain zeros. Alternatively, y can be an svsim object. In this case, the returns will be extracted and a warning is thrown.
draws	single number greater or equal to 1, indicating the number of draws after burn-in (see below). Will be automatically coerced to integer. The default value is 10000.
burnin	single number greater or equal to 0, indicating the number of draws discarded as burn-in. Will be automatically coerced to integer. The default value is 1000.
designmatrix	regression design matrix for modeling the mean. Must have length(y) rows. Alternatively, designmatrix may be a string of the form "arX", where X is a nonnegative integer. To fit a constant mean model, use designmatrix = "ar0" (which is equivalent to designmatrix = matrix(1, nrow = length(y))). To fit an AR(1) model, use designmatrix = "ar1", and so on. If some elements of designmatrix are NA, the mean is fixed to zero (pre-1.2.0 behavior of stochvol).
priormu	numeric vector of length 2, indicating mean and standard deviation for the Gaussian prior distribution of the parameter μ , the level of the log-volatility. The default value is c(0, 100), which constitutes a practically uninformative prior for common exchange rate datasets, stock returns and the like.
priorphi	numeric vector of length 2, indicating the shape parameters for the Beta prior distribution of the transformed parameter $(\phi+1)/2$, where ϕ denotes the persistence of the log-volatility. The default value is c(5, 1.5), which constitutes a prior that puts some belief in a persistent log-volatility but also encompasses the region where ϕ is around 0.

priorsigma	single positive real number, which stands for the scaling of the transformed parameter σ^2 , where σ denotes the volatility of log-volatility. More precisely, $\sigma^2 \sim \text{priorsigma} * \text{chisq}(\text{df} = 1)$. The default value is 1, which constitutes a reasonably vague prior for many common exchange rate datasets, stock returns and the like.
priornu	numeric vector of length 2 (or NA), indicating the lower and upper bounds for the uniform prior distribution of the parameter ν , the degrees-of-freedom parameter of the conditional innovations t-distribution. The default value is NA, fixing the degrees-of-freedom to infinity. This corresponds to conditional standard normal innovations, the pre-1.1.0 behavior of stochvol .
priorbeta	numeric vector of length 2, indicating the mean and standard deviation of the Gaussian prior for the regression parameters. The default value is $c(0, 10000)$, which constitutes a very vague prior for many common datasets. Not used if <code>designmatrix</code> is NA.
thinpara	single number greater or equal to 1, coercible to integer. Every <code>thinpara</code> parameter draw is kept and returned. The default value is 1, corresponding to no thinning of the parameter draws i.e. every draw is stored.
thinlatent	single number greater or equal to 1, coercible to integer. Every <code>thinlatent</code> latent variable draw is kept and returned. The default value is 1, corresponding to no thinning of the latent variable draws, i.e. every draw is kept.
thintime	single number greater or equal to 1, coercible to integer. If <code>thintime</code> is different from 1, only every <code>thintime</code> latent log-volatility is being monitored. If, e.g., <code>thintime = 3</code> , the latent log-volatilities h_1, h_4, h_7, \dots will be kept. The default value is 1, meaning that all latent variables h_1, h_2, h_3, \dots are stored.
keptau	logical value indicating whether the 'variance inflation factors' should be stored (used for the sampler with conditional t innovations only). This may be useful to check at what point(s) in time the normal disturbance had to be 'upscaled' by a mixture factor and when the series behaved 'normally'.
quiet	logical value indicating whether the progress bar and other informative output during sampling should be omitted. The default value is FALSE, implying verbose output.
startpara	<i>optional</i> named list, containing the starting values for the parameter draws. If supplied, <code>startpara</code> must contain three elements named <code>mu</code> , <code>phi</code> , and <code>sigma</code> , where <code>mu</code> is an arbitrary numerical value, <code>phi</code> is a real number between -1 and 1, and <code>sigma</code> is a positive real number. The default value is <code>list(mu = -10, phi = 0.9, sigma = 0.3)</code> .
startlatent	<i>optional</i> vector of length <code>length(x\$y)</code> , containing the starting values for the latent log-volatility draws. The default value is <code>rep(-10, length(x\$y))</code> .
expert	<i>optional</i> named list of expert parameters. For most applications, the default values probably work best. Interested users are referred to the literature provided in the References section. If <code>expert</code> is provided, it may contain the following named elements: <code>parameterization</code> : Character string equal to "centered", "noncentered", "GIS_C", or "GIS_NC". Defaults to "GIS_C". <code>mhcontrol</code> : Single numeric value controlling the proposal density of a Metropolis-Hastings (MH) update step when sampling <code>sigma</code> . If <code>mhcontrol</code> is smaller than

0, an independence proposal will be used, while values greater than zero control the stepsize of a log-random-walk proposal. Defaults to -1.

`gammaprior`: Single logical value indicating whether a Gamma prior for σ^2 should be used. If set to FALSE, an Inverse Gamma prior is employed. Defaults to TRUE.

`truncnormal`: Single logical value indicating whether a truncated Gaussian distribution should be used as proposal for draws of ϕ . If set to FALSE, a regular Gaussian prior is employed and the draw is immediately discarded when values outside the unit ball happen to be drawn. Defaults to FALSE.

`mhsteps`: Either 1, 2, or 3. Indicates the number of blocks used for drawing from the posterior of the parameters. Defaults to 2.

`proposalvar4sigmaphi`: Single positive number indicating the conditional prior variance of $\sigma \cdot \phi$ in the ridge *proposal* density for sampling (μ, ϕ) . Defaults to 10^8 .

`proposalvar4sigmatheta`: Single positive number indicating the conditional prior variance of $\sigma \cdot \theta$ in the ridge *proposal* density for sampling (μ, ϕ) . Defaults to 10^{12} .

... Any extra arguments will be forwarded to [updatesummary](#), controlling the type of statistics calculated for the posterior draws.

Details

For details concerning the algorithm please see the paper by Kastner and Frühwirth-Schnatter (2014).

Value

The value returned is a list object of class `svdraws` holding

<code>para</code>	mcmc object containing the <i>parameter</i> draws from the posterior distribution.
<code>latent</code>	mcmc object containing the <i>latent instantaneous log-volatility</i> draws from the posterior distribution.
<code>latent0</code>	mcmc object containing the <i>latent initial log-volatility</i> draws from the posterior distribution.
<code>tau</code>	mcmc object containing the <i>latent variance inflation factors</i> for the sampler with conditional t-innovations (<i>optional</i>).
<code>beta</code>	mcmc object containing the <i>regression coefficient</i> draws from the posterior distribution (<i>optional</i>).
<code>y</code>	the argument <code>y</code> .
<code>runtime</code>	<code>proc_time</code> object containing the run time of the sampler.
<code>priors</code>	list containing the parameter values of the prior distribution, i.e. the arguments <code>priormu</code> , <code>priorphi</code> , <code>priorsigma</code> , and potentially <code>priornu</code> and <code>priorbeta</code> .
<code>thinning</code>	list containing the thinning parameters, i.e. the arguments <code>thinpara</code> , <code>thinlatent</code> and <code>thintime</code> .
<code>summary</code>	list containing a collection of summary statistics of the posterior draws for <code>para</code> , <code>latent</code> , and <code>latent0</code> .

To display the output, use `print`, `summary` and `plot`. The `print` method simply prints the posterior draws (which is very likely a lot of output); the `summary` method displays the summary statistics currently stored in the object; the `plot` method `plot.svdraws` gives a graphical overview of the posterior distribution by calling `volplot`, `traceplot` and `densplot` and displaying the results on a single page.

Note

If `y` contains zeros, you might want to consider de-meaning your returns or use `designmatrix = "ar0"`.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

References

Kastner, G. and Frühwirth-Schnatter, S. (2014). Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, **76**, 408–423, <http://dx.doi.org/10.1016/j.csda.2013.01.002>.

See Also

`svsim`, `updatesummary`, `predict.svdraws`, `plot.svdraws`.

Examples

```
## Simulate a short and highly persistent SV process
sim <- svsim(100, mu = -10, phi = 0.99, sigma = 0.2)

## Obtain 5000 draws from the sampler (that's not a lot)
draws <- svsample(sim$y, draws = 5000, burnin = 100,
  priormu = c(-10, 1), priorphi = c(20, 1.5), priorsigma = 0.2)

## Check out the results
summary(draws)
plot(draws)

## Another example, this time with an AR(1) structure for the mean
## Not run:
data(exrates)
y <- exrates$USD

## Fit AR(1)-SV model to EUR-USD exchange rates
res <- svsample(y, designmatrix = "ar1")

## Use predict.svdraws to obtain predictive volatilities
ahead <- 100
predvol <- predict(res, steps = ahead)

## Use arpredict to obtain draws from the posterior predictive
preddraws <- arpredict(res, predvol)
```

```

## Calculate predictive quantiles
predquants <- apply(preddraws, 2, quantile, c(.1, .5, .9))

## Visualize
ts.plot(y, xlim = c(length(y) - ahead, length(y) + ahead),
        ylim = range(predquants))
for (i in 1:3) {
  lines((length(y) + 1):(length(y) + ahead), predquants[i,],
        col = 3, lty = i
  )
}

## End(Not run)

```

svsample2

Minimal overhead version of [svsample](#).

Description

svsample2 is a minimal overhead version of [svsample](#) with slightly different default arguments and a simplified return value structure. It is intended to be used mainly for one-step updates where speed is an issue, e.g., as a plug-in into other MCMC samplers. Note that absolutely no input checking is performed, thus this function is to be used with proper care!

Usage

```

svsample2(y, draws = 1, burnin = 0, priormu = c(0, 100),
          priorphi = c(5, 1.5), priorsigma = 1, priornu = NA,
          thinpara = 1, thinlatent = 1, thintime = 1,
          keeptau = FALSE, quiet = TRUE, startpara, startlatent)

```

Arguments

y	numeric vector containing the data (usually log-returns), which must not contain zeroes.
draws	single number greater or equal to 1, indicating the number of draws after burn-in (see below). Will be automatically coerced to integer. The default value is 1.
burnin	single number greater or equal to 0, indicating the number of draws discarded as burn-in. Will be automatically coerced to integer. The default value is 0.
priormu	numeric vector of length 2, indicating mean and standard deviation for the Gaussian prior distribution of the parameter mu, the level of the log-volatility. The default value is $c(0, 100)$, which constitutes a practically uninformative prior for common exchange rate datasets, stock returns and the like.
priorphi	numeric vector of length 2, indicating the shape parameters for the Beta prior distribution of the transformed parameter $(\phi+1)/2$, where phi denotes the persistence of the log-volatility. The default value is $c(5, 1.5)$, which constitutes a prior that puts some belief in a persistent log-volatility but also encompasses the region where phi is around 0.

priorsigma	single positive real number, which stands for the scaling of the transformed parameter σ^2 , where σ denotes the volatility of log-volatility. More precisely, $\sigma^2 \sim \text{priorsigma} * \text{chisq}(\text{df} = 1)$. The default value is 1, which constitutes a reasonably vague prior for many common exchange rate datasets, stock returns and the like.
priornu	numeric vector of length 2 (or NA), indicating the lower and upper bounds for the uniform prior distribution of the parameter ν , the degrees-of-freedom parameter of the conditional innovations t-distribution. The default value is NA, fixing the degrees-of-freedom to infinity. This corresponds to conditional standard normal innovations, the pre-1.1.0 behavior of stochvol .
thinpara	single number greater or equal to 1, coercible to integer. Every thinparath parameter draw is kept and returned. The default value is 1, corresponding to no thinning of the parameter draws – every draw is stored.
thinlatent	single number greater or equal to 1, coercible to integer. Every thinlatentth latent variable draw is kept and returned. The default value is 1, corresponding to no thinning of the latent variable draws, i.e. every draw is kept.
thintime	single number greater or equal to 1, coercible to integer. If thintime is different from 1, only every thintimeth latent log-volatility is being monitored. If, e.g., $\text{thintime} = 3$, the latent log-volatilities h_1, h_4, h_7, \dots will be kept. The default value is 1, meaning that all latent variables h_1, h_2, h_3, \dots are stored.
keeptau	logical value indicating whether the 'variance inflation factors' should be stored (used for the sampler with conditional t innovations only). This may be useful to check at what point(s) in time the normal disturbance had to be 'upscaled' by a mixture factor and when the series behaved 'normally'.
quiet	logical value indicating whether the progress bar and other informative output during sampling should be omitted. The default value is TRUE, implying non-verbose output.
startpara	<i>compulsory</i> named list, containing the starting values for the parameter draws. startpara must contain three elements named mu, phi, and sigma, where mu is an arbitrary numerical value, phi is a real number between -1 and 1, and sigma is a positive real number.
startlatent	<i>compulsory</i> vector of length $\text{length}(x\$y)$, containing the starting values for the latent log-volatility draws.

Details

As opposed to the ordinary [svsample](#), the default values differ for draws, burnin, and quiet. Note that currently neither expert nor ... arguments are provided.

Value

A list with three components:

para	3 times draws matrix containing the parameter draws. If priornu is not NA, this is a 4 times draws matrix.
latent	$\text{length}(y)$ times draws matrix containing draws of the latent variables h_1, \dots, h_n .
latent0	Vector of length draws containing the draw(s) of the initial latent variable h_0 .

Warning

Expert use only! For most applications, the use of [svsample](#) is recommended.

Note

Please refer to the package vignette for an example.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[svsample](#)

svsim

Simulating a Stochastic Volatility Process

Description

svsim is used to produce realizations of a stochastic volatility (SV) process.

Usage

```
svsim(len, mu = -10, phi = 0.98, sigma = 0.2, nu = Inf)
```

Arguments

len	length of the simulated time series.
mu	level of the latent log-volatility AR(1) process. The default value is -10.
phi	persistence of the latent log-volatility AR(1) process. The default value is 0.98.
sigma	volatility of the latent log-volatility AR(1) process. The default value is 0.2.
nu	degrees-of-freedom for the conditional innovations distribution. The default value is Inf, corresponding to standard normal conditional innovations.

Details

This function draws an initial log-volatility h_0 from the stationary distribution of the AR(1) process and iteratively generates h_1, \dots, h_n . Finally, the “log-returns” are simulated from a normal distribution with mean 0 and standard deviation $\exp(h/2)$.

Value

The output is a list object of class `svsim` containing

<code>y</code>	a vector of length <code>len</code> containing the simulated data, usually interpreted as “log-returns”.
<code>vol</code>	a vector of length <code>len</code> containing the simulated instantaneous volatilities $\exp(h_t/2)$.
<code>vol0</code>	the initial volatility $\exp(h_0/2)$, drawn from the stationary distribution of the latent AR(1) process.
<code>para</code>	a named list with three elements <code>mu</code> , <code>phi</code> , <code>sigma</code> (and potentially <code>nu</code>), containing the corresponding arguments.

To display the output use `print`, `summary` and `plot`. The `print` method simply prints the content of the object in a moderately formatted manner. The `summary` method provides some summary statistics (in %), and the `plot` method plots the the simulated ‘log-returns’ `y` along with the corresponding volatilities `vol`.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[svsample](#)

Examples

```
## Simulate a highly persistent SV process of length 500
sim <- svsim(500, phi = 0.99, sigma = 0.1)

print(sim)
summary(sim)
plot(sim)
```

updatesummary

Updating the Summary of MCMC Draws

Description

Creates or updates a summary of an `svdraws` object.

Usage

```
updatesummary(x, quantiles = c(0.05, 0.5, 0.95), esspara = TRUE,
              esslatent = FALSE)
```

Arguments

x	svdraws object.
quantiles	numeric vector of posterior quantiles to be computed. The default is <code>c(0.05, 0.5, 0.95)</code> .
esspara	logical value which indicates whether the effective sample size (ESS) should be calculated for the <i>parameter draws</i> . This is achieved by calling <code>effectiveSize</code> from the coda package. The default is TRUE.
esslatent	logical value which indicates whether the effective sample size (ESS) should be calculated for the <i>latent log-volatility draws</i> . This is achieved by calling <code>effectiveSize</code> from the coda package. The default is FALSE, because this can be quite time-consuming when many latent variables are present.

Details

updatesummary will always calculate the posterior mean and the posterior standard deviation of the raw draws and some common transformations thereof. Moreover, the posterior quantiles, specified by the argument `quantiles`, are computed. If `esspara` and/or `esslatent` are TRUE, the corresponding effective sample size (ESS) will also be included.

Value

The value returned is an updated list object of class `svdraws` holding

para	mcmc object containing the <i>parameter</i> draws from the posterior distribution.
latent	mcmc object containing the <i>latent instantaneous log-volatility</i> draws from the posterior distribution.
latent0	mcmc object containing the <i>latent initial log-volatility</i> draws from the posterior distribution.
y	argument <code>y</code> .
runtime	"proc_time" object containing the run time of the sampler.
priors	list containing the parameter values of the prior distribution, i.e. the arguments <code>priormu</code> , <code>priorphi</code> , <code>priorsigma</code> (and potentially <code>nu</code>).
thinning	list containing the thinning parameters, i.e. the arguments <code>thinpara</code> , <code>thinlatent</code> and <code>thintime</code> .
summary	list containing a collection of summary statistics of the posterior draws for <code>para</code> , <code>latent</code> , and <code>latent0</code> .

To display the output, use `print`, `summary` and `plot`. The `print` method simply prints the posterior draws (which is very likely a lot of output); the `summary` method displays the summary statistics currently stored in the object; the `plot` method gives a graphical overview of the posterior distribution by calling `volplot`, `traceplot` and `densplot` and displaying the results on a single page.

Note

updatesummary does not actually overwrite the object's current summary, but in fact creates a new object with an updated summary. Thus, don't forget to overwrite the old object if this is what you intend to do. See the examples below for more details.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[svsample](#)

Examples

```
## Here is a baby-example to illustrate the idea.
## Simulate an SV time series of length 51 with default parameters:
sim <- svsim(51)

## Draw from the posterior (but save only every fifth point in time):
res <- svsample(sim$, draws = 7000, thintime = 5, priorphi = c(10, 1.5))

## Check out the results:
summary(res)
plot(res)

## Look at other quantiles and calculate ESS of latents:
newquants <- c(0.01, 0.05, 0.25, 0.5, 0.75, 0.95, 0.99)
res <- updatesummary(res, quantiles = newquants, esslatent = TRUE)

## See the difference?
summary(res)
plot(res)
```

volplot

Plotting Quantiles of the Latent Volatilities

Description

Displays quantiles of the posterior distribution of the volatilities over time as well as predictive distributions of future volatilities.

Usage

```
volplot(x, forecast = 0, dates = NULL, show0 = FALSE, col = NULL,
        forecastlty = NULL, tcl = -0.4, mar = c(1.9, 1.9, 1.9, 0.5),
        mgp = c(2, 0.6, 0), simobj = NULL, ...)
```

Arguments

x	svdraws object.
forecast	nonnegative integer or object of class svpredict, as returned by predict.svdraws . If an integer greater than 0 is provided, predict.svdraws is invoked to obtain the forecast-step-ahead prediction. The default value is 0.

dates	vector of length <code>ncol(x\$latent)</code> , providing optional dates for labeling the x-axis. The default value is <code>NULL</code> ; in this case, the axis will be labeled with numbers.
show0	logical value, indicating whether the initial volatility $\exp(h_0/2)$ should be displayed. The default value is <code>FALSE</code> .
col	vector of color values (see par) used for plotting the quantiles. The default value <code>NULL</code> results in gray lines for all quantiles except the median, which is displayed in black.
forecastlty	vector of line type values (see par) used for plotting quantiles of predictive distributions. The default value <code>NULL</code> results in dashed lines.
tcl	The length of tick marks as a fraction of the height of a line of text. See par for details. The default value is <code>-0.4</code> , which results in slightly shorter tick marks than usual.
mar	numerical vector of length 4, indicating the plot margins. See par for details. The default value is <code>c(1.9, 1.9, 1.9, 0.5)</code> , which is slightly smaller than the R-defaults.
mgp	numerical vector of length 3, indicating the axis and label positions. See par for details. The default value is <code>c(2, 0.6, 0)</code> , which is slightly smaller than the R-defaults.
simobj	object of class <code>svsim</code> as returned by the SV simulation function <code>svsim</code> . If provided, “true” data generating values will be added to the plot(s).
...	further arguments are passed on to the invoked <code>ts.plot</code> function.

Value

Called for its side effects. Returns argument `x` invisibly.

Note

In case you want different quantiles to be plotted, use [updatesummary](#) on the `svdraws` object first. An example of doing so is given below.

Author(s)

Gregor Kastner <gregor.kastner@wu.ac.at>

See Also

[updatesummary](#), [paratraceplot](#), [paradensplot](#), [plot.svdraws](#).

Examples

```
## Simulate a short and highly persistent SV process
sim <- svsim(100, mu = -10, phi = 0.99, sigma = 0.2)

## Obtain 5000 draws from the sampler (that's not a lot)
draws <- svsample(sim$y, draws = 5000, burnin = 100,
  priormu = c(-10, 1), priorphi = c(20, 1.5),
```

```
priorsigma = 0.2)  
  
## Plot the latent volatilities and some forecasts  
volplot(draws, forecast = 10)  
  
## Re-plot with different quantiles  
newquants <- c(0.01, 0.05, 0.25, 0.5, 0.75, 0.95, 0.99)  
draws <- updatesummary(draws, quantiles=newquants)  
  
volplot(draws, forecast = 10)
```

Index

- *Topic **datagen**
 - svsim, 18
- *Topic **datasets**
 - exrates, 4
- *Topic **hplot**
 - paradensplot, 6
 - paratraceplot, 8
 - plot.svdraws, 9
 - volplot, 21
- *Topic **models**
 - stochvol-package, 2
 - svsample, 12
 - svsample2, 16
- *Topic **package**
 - stochvol-package, 2
- *Topic **ts**
 - arpredict, 3
 - predict.svdraws, 11
 - stochvol-package, 2
 - svsample, 12
 - svsample2, 16
 - svsim, 18
 - volplot, 21
- *Topic **utilities**
 - Extractors, 5
 - logret, 6
 - updatesummary, 19
 - .svsample (svsample2), 16
- arpredict, 3
- density, 6, 7
- densplot, 7, 15, 20
- effectiveSize, 20
- exrates, 4
- Extractors, 5
- latent (Extractors), 5
- latent0 (Extractors), 5
- logret, 6
- par, 7–9, 22
- para (Extractors), 5
- paradensplot, 6, 8, 10, 22
- paratraceplot, 7, 8, 10, 22
- plot.svdraws, 7, 8, 9, 11, 15, 22
- predict.svdraws, 3, 9, 11, 15, 21
- priors (Extractors), 5
- runtime (Extractors), 5
- stochvol (stochvol-package), 2
- stochvol-package, 2
- svsample, 3, 4, 12, 16–19, 21
- svsample2, 16
- svsim, 7, 8, 10, 15, 18, 22
- thinning (Extractors), 5
- traceplot, 8, 15, 20
- updatesummary, 10, 14, 15, 19, 22
- volplot, 7, 8, 10, 11, 15, 20, 21