Package 'tsbugs'

February 20, 2015

Type Package

Title Create time series BUGS models.

Version 1.2

Author Guy J. Abel

Maintainer ``Guy J. Abel" <g.j.abel@gmail.com>

Depends R(>= 2.15.2)

Suggests R2OpenBUGS, fanplot

Description The tsbugs package contains a collection of R functions that can be used to create time series BUGS models of various order. Included are function to create BUGS with non-constant variance such stochastic volatility models and random variance shift models.

License GPL-2

LazyLoad yes

LazyData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-02-25 15:02:00

R topics documented:

tsbugs-package *Create time series BUGS models.*

Description

The tsbugs package contains a collection of R functions that can be used to create time series BUGS models of various order. Included are function to create BUGS with non-constant variance such stochastic volatility models and random variance shift models.

Details

Package:	tsbugs
Type:	Package
Version:	1.0
License:	GPL-2

Author(s)

Guy J. Abel

Maintainer: Guy J. Abel <g.j.abel@gmail.com>

ar.bugs *Create BUGS Script of a Autoregressive (AR) Time Series Model*

Description

Create BUGS script of an time series model, where the data is assumed to be normally distributed. Options allow for the inclusion of a different lag orders for the mean term, forecasts, posterior simulations from the model and alternative specification of prior distributions on each parameter.

Usage

```
ar.bugs(y, ar.order = 1, k = NULL, sim = FALSE, mean.centre = FALSE, beg = ar.order + 1,
mean.prior = ar.prior, ar.prior = "dnorm(0,1)",
tol.prior = "dgamma(0.000001,0.000001)", var.prior = NULL, sd.prior = NULL,
space = FALSE)
```

Arguments

y	Data to be used for the BUGS model.
ar.order	AR order of the mean process for BUGS model.
k	Length of forecast horizon to be included in the BUGS model.
sim	Enable posterior simulations to be included in the BUGS model. Default is FALSE.
mean.centre	Include a term to centre the data on its mean value.Default is FALSE.
beg	Starting value for which data are considered onwards (and including) in the likelihood of the BUGS model. Default is ar.order+1 but if comparing models of different orders, users may wish to set all beg to the same value.
mean.prior	Prior for mean term (not used if mean.centre is not set to TRUE). The distribution should be stated in BUGS syntax. By default, the same prior as the autoregressive terms is used.
ar.prior	Prior for autoregressive terms. The distribution should be stated in BUGS syntax. By default this is set to a normal distribution with mean 0 and tolerance 1 (dnorm(0,1)). The same prior are used for all autoregressive terms.
tol.prior	Prior for the tolerance of the model. The distribution should be stated in BUGS syntax. By default this is set to a uninformative gamma distribution.
var.prior	Prior for the variance of the model. This must be a distribution of syntax recognisable to BUGS. This is not set by default (where a gamma distribution for the tolerance is used).
sd.prior	Prior for the standard deviation of the model. This must be a distribution of syntax recognisable to BUGS. This is not set by default (where a gamma distribution for the tolerance is used).
space	Include some additional empty lines to seperate the likelihood, priors, forcasts and simulation components of the BUGS model.

Details

This function creates BUGS scripts of an AR time series model. Prior distributions should be set up using BUGS syntax. For example, dnorm(.,.) is a normal distribution with mean and tolerance (not variance) arguments. Only one argument of the tol.prior, var.prior or sd.prior should be set. Also, any new prior for either of these parameters should be based on distributions that provide only positive values, otherwise the BUGS model is likely to fail.

The data, y, can contain missing values. Note, if missing values are close the beginning of the series when a high order model specified (i.e. the second data point is missing and a AR(4) is specified) the user with have to set a high starting point for model to be fitted on (beg) for the BUGS model to function (i.e. 7).

Value

bug	A BUGS model of type tsbugs.
data	The data to be used with the model. This might extend the original data passed to the function with k unknown future values to be forecast.
info	Additional information on the length of the data, variance type and line numbers of certain parts of the BUGS model.

Author(s)

Guy J. Abel

See Also

sv.bugs, rv.bugs

Examples

```
# Create AR(1) model for Lake Huron data
plot(LakeHuron)
LH <- LakeHuron
ar1 <- ar.bugs(y = diff(LH), ar.order = 1)
print(ar1)

# AR(2) model with alternative prior
ar2 <- ar.bugs(y = diff(LH), ar.order = 2, ar.prior = "dunif(-1,1)", var.prior = "dgamma(0.001,0.001)")
print(ar2)

# AR(3) model with forecast and posterior simulations
ar3 <- ar.bugs(y = diff(LH), ar.order = 3, k = 10, sim = TRUE, mean.centre = TRUE)
print(ar3)

## Not run:
# Run in OpenBUGS
writeLines(ar3$bug, "ar3.txt")
library("R2OpenBUGS")

ar3.bug <- bugs(data = ar3$data,
inits = list(inits(ar3)),
param = c(nodes(ar3, "prior")$name, "y.sim", "y.new"),
model = "ar3.txt",
n.iter = 11000, n.burnin = 1000, n.chains = 1)

# Plot the parameters posteriors and traces
library("coda")
param.mcmc <- as.mcmc(ar3.bug$sims.matrix[, nodes(ar3, "prior")$name])
plot(param.mcmc)

# Plot posterior simulations using fanplot
library("fanplot")
y.mcmc <- ar3.bug$sims.list$y.sim
y.pn <- pn(y.mcmc, st = tsp(diff(LH))[1] + 1)
plot(diff(LH), type = "n")
fan(y.pn)
lines(diff(LH))

## End(Not run)
```

ew *Population Data for England and Wales*

Description

Population data for England and Wales combined between 1841 and 2007

Usage

```
data(ew)
```

Format

The format is: Time-Series [1:167] from 1841 to 2007: 15845661 16021370 16201588 16399953 16606433 ...

Source

The Human Mortality Database (http://www.mortality.org).

References

Human Mortality Database (2012). University of California, Berkeley (USA) and Max Planck Institute for Demographic Research (Germany).

Examples

```
data(ew)
plot(ew)
```

inits *Produce a Set of Candidate Initial Values*

Description

Creates a list of initial values for a tsbugs object to help shorten your code when running models through R2WinBUGS or R2OpenBUGS.

Usage

```
inits(bug, warn.mess = TRUE)
```

Arguments

bug	A time series BUGS model created using the tsbugs package.
warn.mess	Print warning message notifying users that the initial values simple guesses and have nothing to do with data. Default is TRUE.

Details

The `inits` function is intended to provide a set of candidate initial values for BUGS models created using the `tsbugs` package. The list provided from `inits` does not guarantee that BUGS model will run from an efficient set of starting points, or indeed that the BUGS model will run at all. Values are set to be in the parameter space and are not based on the data used to create the `tsbugs` model. If problems do occur running the BUGS model, users are advised to set their own initial values.

Value

`list` with components dependent on model type.

Author(s)

Guy J. Abel

See Also

`ar.bugs`, `sv.bugs`, `rv.bugs`

Examples

```
# Create AR(4) model for Lake Huron data
LH <- LakeHuron
ar4 <- ar.bugs(y = diff(LH), ar.order = 4)

# Candidate initial values
inits(ar4)
```

---

nodes *Data Frame of Nodes within a BUGS model.*

Description

Provides a `data.frame` of both random and deterministic nodes in a `tsbugs` BUGS model. Can also be used to extract nodes from just a single part of the model, such as the prior distributions. This might be of particular use to users when setting up parameters to follow when running models through R2WinBUGS or R2OpenBUGS.

Usage

```
nodes(bug, part = NULL)
```

Arguments

`bug`	A time series BUGS model created using the `tsbugs` package.
`part`	A part of the `tsbugs` models. Must be one of `"likelihood"`, `"priors"`, `"forecast"` or `"simulation"`. Abbreviations of these parts are allowed (`pmatch` is used for the matching).

Details

Provides a summary of nodes in all of, or part of, the tsbugs BUGS model. Returns a data frame with details on each node. When nodes are within a loop additional details on the starting and ending time points for the node are also given.

Value

A data.frame with columns:

name	Name of node.
type	Stochastic node represented by ~. Deterministic node represented by <-.
dt	Distribution or transformation in node.
beg	Beginning of node loop.
end	End of node loop.
stoc	Binary representation of type column, where 1 represents a stochastic node and 0 a deterministic node.
id	Line number of node in the BUGS model.
dist	Distribution in node.
param1	First parameter of distribution.
param2	Second parameter of distribution.

Author(s)

Guy J. Abel

Examples

```
# Create AR(1) model for Lake Huron data
LH <- LakeHuron
ar1 <- ar.bugs(y = diff(LH), ar.order = 1)

# All nodes in model
nodes(ar1)

# Priors in model
nodes(ar1, "priors")
```

---

print.tsbugs *Prints tsbugs object*

Description

Prints the BUGS model in a tsbugs object.

Usage

```
## S3 method for class 'tsbugs'
print(x, ...)
```

Arguments

x	A BUGS model of class tsbugs
...	Additional arguments passed to print

Details

Prints tsbugs BUGS model. Note, in a tsbugs object there are additional details (the data used and other summary information) that are not printed.

Author(s)

Guy J. Abel

Examples

```
# Create AR(1) model for Lake Huron data
LH <- LakeHuron
ar1 <- ar.bugs(y = diff(LH), ar.order = 1)

print(ar1)
```

---

rv.bugs *Create BUGS Script of a Random Variance Shift Model*

Description

Create BUGS script of an Random Variance (RV) shift model similar to that of McCulloch and Tsay (1991). Options allow for the inclusion of a different lag orders for the mean term, forecasts, posterior simulations from the model and alternative specification of prior distributions on some parameters.

Usage

```
rv.bugs(y, ar.order = 0, k = NULL, sim = FALSE, mean.centre = FALSE, beg = ar.order + 1,
mean.prior = ar.prior, ar.prior = "dnorm(0,1)",
rv.tol0.prior = "dgamma(0.000001,0.000001)", rv.eps.prior = "dbeta(1, 100)",
rv.ilambda2.prior = "dgamma(0.01,0.01)",
space = FALSE)
```

Arguments

y	Data to be used for the BUGS model.
ar.order	AR order of the mean process for BUGS model.
k	Length of forecast horizon to be included in the BUGS model.
sim	Enable posterior simulations to be included in the BUGS model. Default is FALSE.
mean.centre	Include a term to centre the data on its mean value. Default is FALSE.
beg	Starting value for which data are considered onwards (and including) in the likelihood of the BUGS model. Default is ar.order+1 but if comparing models of different orders, users may wish to set all beg to the same value.
mean.prior	Prior for mean term (not used if mean.centre is not set to TRUE). The distribution should be stated in BUGS syntax. By default, the same prior as the autoregressive terms are used.
ar.prior	Prior for autoregressive terms. The distribution should be stated in BUGS syntax. By default this is set to a normal distribution with mean 0 and tolerance 1 (dnorm(0,1)). The same prior is used for all autoregressive terms.
rv.tol0.prior	Prior distribution for the time-specific tolerance of the first data point, from which potential future shifts are based on. This must be a distribution of syntax recognisable to BUGS. By default this is set to a uninformative gamma distribution.
rv.eps.prior	Prior distribution for the epsilon term (the probability of the variance shift). This must be a distribution of syntax recognisable to BUGS and in theory restricted to generate values between 0 and 1, although no check is made to ensure users specify distributions as such. By default this argument is set to a beta distribution with a small probability for a variance shift (dbeta(1,100)).
rv.ilambda2.prior	Prior for the inverse of the squared lambda term of a variance shift model. The lambda represents the magnitude (on the log scale) of average variance shifts, thus ilambda2 in a BUGS model is a similar to a tolerance on the shifts in the variance, and hence must be positive. By default this is set to a gamma distribution (dgamma(0.01,0.01)).
space	Include some additional empty lines to seperate the likelihood, priors, forcasts and simulation components of the BUGS model.

Details

This function create BUGS scripts of an random variance shift model adapted from McCulloch and Tsay (1991). Prior distributions for the tolerance of the initial data point, the probability of the variance shift, and the magnitude of average variance shifts can be specified. by users to differ from their default values. User specified prior distributions should be set up using BUGS syntax. For example, dnorm is a normal distribution with mean and tolerance (not variance) arguments.

The data y, can contain missing values. Note, if missing values are close the beginning of the series when a high order model for the mean process is specified (i.e. the second data point is missing and a AR(4) is specified) the user with have to set a high starting point for model to be fitted on (beg) for the BUGS model to function (i.e. 7).

Value

bug	A BUGS model of type tsbugs.
data	The data to be used with the model. This might extend the original data passed to the function with k unknown future values to be forecast.
info	Additional information on the length of the data, variance type and line numbers of certain parts of the BUGS model.

Author(s)

Guy J. Abel

References

McCulloch, R. E. and R. S. Tsay (1993). Bayesian Inference and Prediction for Mean and Variance Shifts in Autoregressive Time Series. *Journal of the American Statistical Association* 88 (423), 968–978.

See Also

ar.bugs, sv.bugs

Examples

```
# Create AR(0)-SV model for population growth rate
r <- ts(ew[2:167]/ew[1:166]-1, start=1841)
y <- diff(r)
plot(y, type = "l")
rv0 <- rv.bugs(y)
print(rv0)

# AR(2)-RV model with alternative priors
rv2 <- rv.bugs(y, rv.eps.prior = "dbeta(1,20)")
print(rv2)

# AR(0)-RV model with posterior simulations
rv0 <- rv.bugs(y, sim = TRUE)
print(rv0)

## Not run:
  # Run in OpenBUGS
  writeLines(rv0$bug, "rv0.txt")
  library("R2OpenBUGS")

  # Run model (can take some time, depending on data length)
  rv0.bug <- bugs(data = rv0$data,
    inits = list(inits(rv0)),
    param = c(nodes(rv0, "prior")$name, "y.sim", "y.new"),
    model = "rv0.txt",
    n.iter = 11000, n.burnin = 1000, n.chains = 1)

  # Plot the parameters posteriors and traces
```

```
library("coda")
param.mcmc <- as.mcmc(rv0.bug$sims.matrix[, nodes(rv0, "prior")$name])
plot(param.mcmc)

# Plot posterior simulations using fanplot
library("fanplot")
y.mcmc <- rv0.bug$sims.list$y.sim
y.pn <- pn(y.mcmc, st = start(y)[1])
plot(y, type = "n")
fan(y.pn)
lines(y)

# Plot volatility
h.mcmc <- rv0.bug$sims.list$h
h.pn <- pn(h.mcmc, st = start(y)[1])
sigma.pn <- pn(sims = sqrt(exp(h.mcmc)), st = start(y)[1])
par(mfrow = c(2, 1), mar = rep(2, 4))
plot(NULL, type = "n", xlim = tsp(h.pn)[1:2], ylim = range(h.pn[, 5:95]), main = "h_t")
fan(h.pn)
plot(NULL, type = "n", xlim = tsp(sigma.pn)[1:2], ylim = range(sigma.pn[, 1:95]), main = "sigma_t")
fan(sigma.pn)

## End(Not run)
```

`sv.bugs` *Create BUGS Script of a Stochastic Volatility (SV) Model*

Description

Create BUGS script of an SV time series model, similar those in Meyer and Yu (2002). Options allow for the inclusion of a different lag orders for the mean term, forecasts, posterior simulations from the model and alternative specification of prior distributions on each parameters.

Usage

```
sv.bugs(y, ar.order = 0, k = NULL, sim = FALSE, mean.centre = FALSE, beg = ar.order + 1,
mean.prior = ar.prior, ar.prior = "dnorm(0,1)",
sv.order = 1,
sv.mean.prior1 = "dnorm(0,0.001)", sv.mean.prior2 = NULL,
sv.ar.prior1 = "dunif(0,1)", sv.ar.prior2 = NULL,
sv.tol.prior = "dgamma(0.01,0.01)",
space = FALSE)
```

Arguments

`y`	Data to be used for the BUGS model.
`ar.order`	AR order of the mean process for BUGS model.
`k`	Length of forecast horizon to be included in the BUGS model.

sim	Enable posterior simulations to be included in the BUGS model. Default is FALSE.
mean.centre	Include a term to centre the data on its mean value. Default is FALSE.
beg	Starting value for which data are considered onwards (and including) in the likelihood of the BUGS model. By default this is the ar.order+1 but if comparing models of different orders, users may wish to set all beg to the same value.
mean.prior	Prior for mean term (not used if mean.centre is not set to TRUE). The distribution should be stated in BUGS syntax. By default, the same prior as the autoregressive terms are used.
ar.prior	Prior for autoregressive terms. The distribution should be stated in BUGS syntax. By default set to a normal distribution with mean 0 and tolerance 1 (dnorm(0,1)). The same prior is used for all autoregressive terms.
sv.order	AR order of the volatility process for BUGS model.
sv.mean.prior1	Prior distribution for the mean volatility term. The distribution should be stated in BUGS syntax. By default set to a normal distribution with mean 0 and tolerance 0.001 (dnorm(0,0.001)), as in Meyer and Yu (2002).
sv.mean.prior2	Alternative prior for mean volatility term. The distribution set here will be transformed by taking the negative logarithm. The distribution should be stated in BUGS syntax.
sv.ar.prior1	Prior for autoregressive terms of the volatility process. The distribution set here will be transformed by doubling and then subtracting 1 as in Meyer and Yu (2002). The distribution syntax must be recognisable to BUGS. By default set to a uniform distribution with bounds 0 and 1 (dunif(0,1)).
sv.ar.prior2	Alternative prior for autoregressive terms of the volatility process. The distribution should be stated in BUGS syntax.
sv.tol.prior	Prior for the tolerance of the volatility. The distribution should be stated in BUGS syntax. By default set to a uninformative gamma distribution.
space	Include some additional empty lines to seperate the likelihood, priors, forcasts and simulation components of the BUGS model.

Details

This function create BUGS scripts of an SV time series model, similar those in Meyer and Yu (2002). Prior distributions should be set up using BUGS syntax. For example, dnorm is a normal distribution with mean and tolerance (not variance) arguments. Prior distributions for the mean and autoregressive parameter in the volatility process can be specified directly in sv.mean.prior1 and sv.ar.prior1 respectively. In sv.mean.prior2 the prior is given on the mean variance, and then transformed to the mean volatility. In sv.ar.prior2 the prior is given on the positive scale as illustrated in Meyer and Yu (2002). Only one set of prior distributions are allowed (i.e. either sv.ar.prior1 or sv.ar.prior2 should specified by the user).

The data y, can contain missing values. Note, if missing values are close the beginning of the series when a high order model for the mean process is specified (i.e. the second data point is missing and a AR(4) is specified) the user with have to set a high starting point for model to be fitted on (beg) for the BUGS model to function (i.e. 7).

Value

bug	A BUGS model of type tsbugs.
data	The data to be used with the model. This might extend the original data passed to the function with k unknown future values to be forecast.
info	Additional information on the length of the data, variance type and line numbers of certain parts of the BUGS model.

Author(s)

Guy J. Abel

References

Meyer, R. and J. Yu (2002). BUGS for a Bayesian analysis of stochastic volatility models. *Econometrics Journal* 3 (2), 198–215.

See Also

ar.bugs, rv.bugs

Examples

```
# Create AR(0)-SV model for svpdx
y <- svpdx$pdx
plot(y, type = "l")
sv0 <- sv.bugs(y)
print(sv0)

# AR(1)-SV model with AR(2) structure in the volatility
sv1 <- sv.bugs(y, ar.order = 1, sv.order = 2)
print(sv1)

# AR(0)-SV model with alternative prior
sv0 <- sv.bugs(y, sv.ar.prior2 = "dunif(-1,1)")
print(sv0)

# AR(0)-SV model with forecast and posterior simulations
sv0 <- sv.bugs(y, k = 10, sim = TRUE)
print(sv0)

## Not run:
  # Run in OpenBUGS
  writeLines(sv0$bug, "sv0.txt")
  library("R2OpenBUGS")

  sv0.bug <- bugs(data = sv0$data,
     inits = list(inits(sv0)),
     param = c(nodes(sv0, "prior")$name, "y.sim", "y.new"),
     model = "sv0.txt",
     n.iter = 20000, n.burnin = 10000, n.chains = 1)
```

```
# Plot the parameters posteriors and traces
library("coda")
param.mcmc <- as.mcmc(sv0.bug$sims.matrix[, nodes(sv0, "prior")$name])
plot(param.mcmc)

# Plot posterior simulations using fanplot
library("fanplot")
y.mcmc <- sv0.bug$sims.list$y.sim
y.pn <- pn(y.mcmc, st = 1)
plot(y, type = "n")
fan(y.pn)
lines(y)

# Plot forecast using fanplot
ynew.mcmc <- sv0.bug$sims.list$y.new
ynew.pn <- pn(ynew.mcmc, st = sv0$info$n + 1)
plot(y, type = "n", xlim = sv0$info$n + c(-100, 20))
fan(ynew.pn)
lines(y)

# Plot volatility
h.mcmc <- sv0.bug$sims.list$h
h.pn <- pn(h.mcmc, st = 1)
sigma.pn <- pn(sims = sqrt(exp(h.mcmc)), st = 1)
par(mfrow = c(2, 1), mar = rep(2, 4))
plot(NULL, type = "n", xlim = tsp(h.pn)[1:2], ylim = range(h.pn[, 5:95]), main = "h_t")
fan(h.pn)
abline(v = length(y))
plot(NULL, type = "n", xlim = tsp(sigma.pn)[1:2], ylim = range(sigma.pn[, 1:95]), main = "sigma_t")
fan(sigma.pn)
abline(v = length(y))

## End(Not run)
```

---

svpdx *Pound-Dollar Exchange Rate Data*

Description

Pound-Dollar exchange rate data from 2nd October 1981 to 28th June 1985.

Usage

```
data(svpdx)
```

Format

A data frame with 945 observations on the following 2 variables.

`date` Date of observation.

`pdx` Logarithm of returns for Pound-Dollar exchange.

Details

Raw data on log returns.

Source

http://www.econ.vu.nl/koopman/sv/svpdx.dat

References

Meyer, R. and J. Yu (2002). BUGS for a Bayesian analysis of stochastic volatility models. *Econometrics Journal* 3 (2), 198–215.

Examples

```
data(svpdx)

# plot
plot(svpdx$pdx, type = "l", xaxt = "n", xlab = "Time", ylab = "Return")

# add x-axis
svpdx$rdate<-format(svpdx$date, format = "%b %Y")
mth <- unique(svpdx$rdate)
qtr <- mth[seq(1,length(mth),3)]
axis(1, at = match(qtr, svpdx$rdate), labels = qtr, cex.axis = 0.75)
axis(1, at = match(mth, svpdx$rdate), labels = FALSE, tcl = -0.2)
```

Index