Package 'NGSSEML'

August 18, 2019

Type Package

Title Non-Gaussian state space with exact marginal likelihood

Version 1.0

Date 2019-03-08

Author T.R. Santos, D. Gamerman, G.C. Franco

Maintainer Thiago Rezende dos Santos <thiagords@est.ufmg.br>

Description This package provides some functions for modeling and forecasting non-Gaussian time series and reliability data via non-Gaussian state space models with exact marginal likelihood.

Depends R (>= 1.9.0), akima

Imports mvtnorm (>= 0.9-9),fields,compiler,dlm,car

License GPL (>=2)

R topics documented:

FilteringF *Filtering and One-Step-Ahead Distributions of the Latent States*

Description

The function FilteringF provides the shape and scale parameters of the filtering and the one-step-ahead forecast distributions of the latent states.

Usage

```
FilteringF(formula,data,na.action="na.omit",pz=NULL,
nBreaks=NULL,model="Poisson",StaPar=NULL,a0=0.01,b0=0.01,amp=FALSE,
distl="PRED",splot=FALSE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`amp`	the interval width is taken in account in the estimation of parameter w which controls the loss of information over time, only for the PEM. For more details see Santos et al. (2017). Default: FALSE. Optional argument.
`distl`	the latent states distribution to be returned.
`splot`	a plot with the point and interval estimates of the states is provided. Optional argument.

Details

Typical usages are

```
FilteringF(Yt~1,data=data.frame(Yt),StaPar=Par,model="Poisson",
a0=0.01,b0=0.01,splot=TRUE)
```

Value

att	'att' is the shape parameter of the one-step-ahead forecast distribution of the states.
btt	'btt' is the scale parameter of the one-step-ahead forecast distribution of the states.
at	'at' is the shape parameter of the filtering distribution of the states. It is necessary to specify this option in the argument 'distl'.
bt	'bt' is the scale parameter of the filtering distribution of the states. It is necessary to specify this option in the argument 'distl'.

Note

It is necessary to specify the argument 'distl' in order to obtain the filtering distribution of the states. The model options are the Poisson, Normal, Laplace, GED, Gamma, Weibull and Generalized Gamma models. 'Zt' are the explanatory time series only for the Normal, Laplace and GED volatility models.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

LikeF SmoothingF

Examples

```
#library(NGSSEML)
Yt=c(1,2,1,4,3)
Par=c(0.9) #w
predpar=FilteringF(Yt~1,data=data.frame(Yt),StaPar=Par,model="Poisson",
a0=0.01,b0=0.01,splot=FALSE)
predpar
filpar=FilteringF(Yt~1,data=data.frame(Yt),StaPar=Par,model="Poisson",
a0=0.01,b0=0.01,distl="FILTER",splot=FALSE)
filpar
```

`gridfunction` *A grid of points for obtaining the static parameters of the non-Gaussian state space models with exact marginal likelihood*

Description

The function builds a grid of the points for the static parameters of the model.

Usage

```
gridfunction(npoints,linf,lsup)
```

Arguments

`npoints`	the number of points/parts that the specified interval of the static parameters is partitioned.
`linf`	the lower limit of the static parameters in the grid.
`lsup`	the upper limit of the static parameters in the grid.

Details

Typical usages are

```
gridfunction(npoints,linf,lsup)
```

Value

`[[1]]`	This function returns the grid of points for the static parameters of the model.

Note

The function is used to perform Bayesian inference for the static parameters. It computes the exact posterior distribution and draws a sample of the marginal posterior distribution of the static parameters using multinomial sampling scheme. It requires the R package 'fields'.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

`ngssm.bayes`

Examples

```
#require("fields")
#library(NGSSEML)
n=5   # number of points
linf=c(0,3,-1)   # lower limit
lsup=c(1,6,-2)     # upper limit
out=gridfunction(n,linf,lsup) # Calling the function
out
```

GridP *Grid function for the PEM*

Description

The function GridP returns a vector with the interval limits (breaks) for the baseline failure rate of the PEM.

Usage

```
GridP(Yt, Event, nT = NULL)
```

Arguments

`Yt`	is a vector with the failure/censored times.
`Event`	is a censoring indicator (a vector).
`nT`	is the number of intervals/breaks. Optional argument. If nT==NULL, it is built one interval per failure is built.

Details

Typical usages are

```
GridP(Yt, Event, nT = NULL)
```

Value

`Break`	a vector with the interval limits (breaks).

Note

If the argument 'nT' is 'NULL', it is built one interval per failure.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

LikeF SmoothingF

Examples

```
#library(NGSSEML)
data(gte_data)
Yt=gte_data$V1
Xt=NULL
Zt=NULL
model="PEM"
amp=FALSE
Event=gte_data$V2
Break=GridP(Yt, Event, nT = NULL)
Break
```

---

gte_data *Daily failure times of 125 telecommunication systems installed by the GTE*

Description

The data are daily failure times of 125 telecommunication systems, including their respective censoring indicator, installed by the GTE corporation in a pre-specified time period (Kim and Proschan 1991).

Usage

```
data(gte_data)
```

Format

A data frame with 125 rows and 2 variables.

Details

The first column of the object gte_data corresponds to the failure times and the second to the censoring indicator.

Source

Kim, J. S. and Proschan, R. (1991). Piecewise exponential estimator of survivor function. IEEE Transactions on Reliability, 40, 134 to 139.

References

Kim, J. S. and Proschan, R. (1991). Piecewise exponential estimator of survivor function. IEEE Transactions on Reliability, 40, 134 to 139.

Examples

```
data(gte_data)
```

LikeF *Marginal Likelihood Function*

Description

This function computes the marginal likelihood function of the static parameters of the model.

Usage

```
LikeF(formula, data,na.action="na.omit",pz=NULL,
nBreaks=NULL,model="Poisson",StaPar=NULL,a0=0.01,b0=0.01,amp=FALSE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`amp`	the interval width is taken in account in the estimation of parameter w which controls the loss of information over time, only for the PEM. For more details see Santos et al. (2017). Default: FALSE. Optional argument.

Details

Typical usages are

```
LikeF2(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model="Poisson",
StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01),a0=0.01,b0=0.01)
```

Value

`llik` This function returns the value of the marginal likelihood function in the logarithmic scale multiplied by -1.

Note

The model options are the Poisson, Normal, Laplace, GED, Gamma, Weibull and Generalized Gamma models. 'Zt' are the explanatory time series only for the Normal, Laplace and GED volatility models.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

FilteringF SmoothingF

Examples

```
#library(NGSSEML)
data(Yt)
Xtm=Yt[,3:7]    # Xt as matrix always!
Xtm[,1]=(1:168-73)/168
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.8,-0.1,0.01,0.01,0.01,0.01)
data1=data.frame(Ytm,Xtm)
LikeF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model="Poisson",StaPar=StaPar,a0=0.01,b0=0.01,amp=FALSE)
```

LikeF2 *Auxiliar Marginal Likelihood Function*

Description

This function is an auxiliar function of the package that computes the marginal likelihood function of the transformed static parameters of the model. All transformed static parameters have values in the interval (-Inf, Inf). The marginal likelihood function is the 'LikeF' function.

Usage

```
LikeF2(formula, data,na.action="na.omit",pz=NULL,
nBreaks=NULL,model="Poisson",StaPar=NULL,a0=0.01,b0=0.01,amp=FALSE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`amp`	the interval width is taken in account in the estimation of parameter w which controls the loss of information over time, only for the PEM. For more details see Santos et al. (2017). Default: FALSE. Optional argument.

Details

Typical usages are

```
LikeF2(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model="Poisson",
StaPar=c(log(-log(0.8)),-0.1,0.01,0.01,0.01,0.01),a0=0.01,b0=0.01)
```

Value

`llik`	This function returns the value of the marginal likelihood function in the logarithmic scale multiplied by -1.

Note

The model options are the Poisson, Normal, Laplace, GED, Gamma, Weibull and Generalized Gamma models. 'Zt' are the explanatory time series only for the Normal, Laplace and GED volatility models.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

LikeF FilteringF SmoothingF

Examples

```
#library(NGSSEML)
data(Yt)
Xtm=as.matrix(Yt[,3:7])    # Xt as matrix always!
Xtm[,1]=Xtm[,1]*1000
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(log(-log(0.8)),-0.1,0.01,0.01,0.01,0.01) # an initial value for the transformed static parameters
LikeF2(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model="Poisson",
StaPar=c(log(-log(0.8)),-0.1,0.01,0.01,0.01,0.01),a0=0.01,b0=0.01)
```

---

ngssm.bayes	*Bayesian estimation of the non-Gaussian state space models with exact marginal likelihood*

Description

The function performs the Bayesian estimation for the static parameters of the model.

Usage

```
ngssm.bayes(formula,data,na.action="na.omit",pz=NULL,nBreaks=NULL,
model="Poisson",StaPar=NULL,amp=FALSE,a0=0.01,b0=0.01,prw=c(1,1),
prnu=NULL,prchi=NULL,prmu=NULL,prbetamu=NULL,prbetasigma=NULL,lower=NULL,
upper=NULL,ci=0.95,pointss=10,nsamplex=1000,postplot=FALSE,contourplot=FALSE,
LabelParTheta=NULL)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event. The value 1 indicates failure.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.
`amp`	the interval width is taken in account in the estimation of parameter w which controls the loss of information over time, only for the PEM. For more details see Santos et al. (2017). Default: FALSE. Optional argument.
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`prw`	a numeric vector of length 2, indicating the hyperparameters of the Beta prior distribution for the parameter w. Optional argument. The default value is c(1,1), which constitutes an uninformative prior for common data sets.
`prnu`	a numeric vector of length 2, indicating the hyperparameters of the Gamma prior distribution for the shape parameter nu. Optional argument.
`prchi`	a numeric vector of length 2, indicating the hyperparameters of the Gamma prior distribution for the shape parameter chi. Optional argument.
`prmu`	a numeric vector of length 2, indicating mean and standard deviation for the Gaussian prior distribution for the parameter mu. Optional argument. This prior can be used in Normal, Laplace and GED time series models.
`prbetamu`	a numeric vector of length p, indicating mean for the Gaussian prior distribution for the parameter beta, the regression coefficients. Optional argument.
`prbetasigma`	a numeric matrix of order p by p, indicating variance-covariance matrix of the Gaussian prior distribution for the parameter beta, the regression coefficients. Optional argument.
`lower`	an lower bound for the static parameters (StaPar) in the density support argument of the ARMS function (MCMC). Optional argument.

upper	an upper bound for the static parameters (StaPar) in the density support argument of the ARMS function (MCMC). Optional argument.
ci	the nominal level of credibility interval for the parameters. Default: ci=0.95. Optional argument.
pointss	the number of points/parts/breaks that the specified interval of the static parameters is partitioned. Default: pointss=10.
nsamplex	the number of samples of the posterior distribution of the static parameters. Default: samples=3000.
postplot	If true, a graph with the marginal posterior distribution of the static parameters is provided. Optional argument.
contourplot	If true, a countour plot of the posterior distribution of the static parameters is provided. Optional argument.
LabelParTheta	If not NULL, the static parameters are called by the specified label. The default value is NULL. Optional argument.

Details

Typical usages are

```
ngssm.bayes(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01),
prw=c(1,1),prbetamu=rep(0,5),prbetasigma=diag(10, 5, 5),pointss=5,nsamplex=1000)
```

Value

[[1]]	This function returns the output of Bayesian estimation for the static parameters.
[[2]]	This function returns posterior samples of the static parameters using multinomial sampling scheme.

Note

This function provides summaries of the posterior distribution of the static parameters of the specified model. In an exact way, the posterior is built to make inferences for the static parameters, and samples of it are drawn using multinomial sampling. If the dimensionality of static parameters and the break number of the grid are high, there are many points to evaluate the posterior distribution and, hence, an MCMC method (ARMS) is used to sample the posterior distribution of the static parameters. Furthermore, it is necessary to specify the limits of the parametric space of the model for the ARMS function in the arguments 'lower' and 'upper'.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

PriorF SmoothingF ngssm.mle

Examples

```
############################################################
##
## Poisson Example: the Polio data
##
############################################################
require(mvtnorm)
require(fields)
require(akima)
require(compiler)
#library(NGSSEML)
#Note: Please, you must check if the packages above have installed correctly.
#Otherwise, NGSSEML package may return some error message.
#### Defaults values (NULL):
#### Inputs:
data(Yt)
Xtm=Yt[,3:7]
Xtm[,1]=(1:168-73)/168
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
#LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01)
#Fit:
a0=0.01
b0=0.01
pointss=8     ### points
#pointss=5     ### points
nsamplex=4000  ##sample
ci=0.95       ## Cred. level
#Fit:
#Bayesian:
set.seed(24666)
fitbayes=ngssm.bayes(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar,
a0=a0,b0=b0,prw=c(1,1),prbetamu=rep(0,5),prbetasigma=diag(10, 5, 5),
ci=ci,pointss=pointss,nsamplex=nsamplex,postplot=FALSE,contourplot=FALSE)
############################################################
##
## GED Example: the PETR3 data
##
############################################################
#GED without Covariate:
#library(NGSSEML)
#### Inputs:
data(Rt)
Ytm=Rt$Rt
Date=Rt$Date
Xtm=NULL
Ztm=NULL
model="GED"
```

```
#LabelParTheta=c("W","nu")
StaPar=c(0.9,1)
p=length(StaPar)
nn=length(Yt)
a0=0.01
b0=0.01
#pointss=500    ### points
pointss=10    ### points

nsamplex=1000 #sample
ci=0.95        # Cred. level
#Bayesian:
fitbayes=ngssm.bayes(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
StaPar=StaPar,a0=a0,b0=b0,prw=c(1,1),prnu=c(0.01,0.01),ci=ci,
pointss=pointss,nsamplex=nsamplex,postplot=FALSE,contourplot=FALSE)

################################################################################
#GED with Covariate:
#### Defaults values (NULL):
data(Rt)
Ytm=Rt$Rt
Date=Rt$Date
Ztm=NULL
nn=length(Ytm)
Ztm=rep(1,nn)
#Xtm=rep(1,nn) # Zt as matrix always!
Xtm=NULL
model="GED"
LabelParTheta=c("W","nu","Mu")
StaPar=c(0.95,1,0)
fitbayes=ngssm.bayes(Ytm~Ztm,data=data.frame(Ytm,Ztm),model=model,pz=1,
StaPar=StaPar,a0=a0,b0=b0,prw=c(1,1),prnu=c(0.01,0.01),prmu=c(0,10),ci=ci,
pointss=pointss,nsamplex=nsamplex,postplot=FALSE,contourplot=FALSE,
LabelParTheta=LabelParTheta)

################################################################################
##
## PEM Example: the GTE data
##
################################################################################
#library(NGSSEML)
#### Inputs:
data(gte_data)
Ytm=gte_data$V1
Eventm=gte_data$V2   # Event: failure, 1.
Breakm=GridP(Ytm, Eventm, nT = NULL)
Xtm=NULL
Ztm=NULL
model="PEM"
amp=FALSE
#LabelParTheta=c("w")
StaPar=c(0.5)
p=length(StaPar)
nn=length(Yt)
a0=0.01
b0=0.01
#pointss=500000    ### points
```

```
pointss=50    ### points
nsamplex=3000 ## Sampling posterior
ci=0.95
alpha=1-ci
#Fit:
fitbayes=ngssm.bayes(Ytm~1,data=data.frame(Ytm,Eventm),model=model,pz=NULL,
StaPar=StaPar,amp=amp,a0=a0,b0=b0,prw=c(1,1),prnu=NULL,prchi=NULL,prmu=NULL,
prbetamu=NULL,prbetasigma=NULL,ci=ci,pointss=pointss,nsamplex=nsamplex,
postplot=FALSE,contourplot=FALSE)

################################################################################
##
##  SR WEIBULL MODEL: the SYS1 data
##
################################################################################
#library(NGSSEML)
#### Defaults values (NULL):
#### Inputs:
data(sys1_data)
Ytm=sys1_data[,1]+0.00001
Xtm=sys1_data[,2]
model="SRWeibull"
#LabelParTheta=c("w","nu","Beta")
pointss=20    ### points
#pointss=50    ### points
#Fit:
StaPar=c(0.98,0.75,0.02)
fitbayes=ngssm.bayes(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=StaPar,
prw=c(1,1),prnu=c(0.1,0.1),prbetamu=rep(0,1),prbetasigma=diag(100,1,1),
pointss=pointss,nsamplex=3000,postplot=FALSE,contourplot=FALSE)

################################################################################
##
##  SR GAMMA MODEL: the SYS1 data
##
################################################################################
#library(NGSSEML)
#### Defaults values (NULL):
#### Inputs:
data(sys1_data)
Ytm=sys1_data[,1]+0.00001
Xtm=sys1_data[,2]
Ztm=NULL
model="SRGamma"
#LabelParTheta=c("w","alpha","Beta")
StaPar=c(0.9,0.7,0.01)
#Fit:
fitbayes=ngssm.bayes(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=StaPar,
prw=c(1,1),prnu=c(0.1,0.1),prbetamu=rep(0,1),prbetasigma=diag(100,1,1),
pointss=pointss,nsamplex=3000,postplot=FALSE,contourplot=FALSE)

################################################################################
################################################################################
```

ngssm.mle — *Maximum likelihood estimation of the non-Gaussian state space models with exact marginal likelihood*

Description

The function performs the marginal likelihood estimation for the static parameters of the model.

Usage

```
ngssm.mle(formula, data,na.action="na.omit",pz=NULL,
nBreaks=NULL,model="Poisson",StaPar=NULL,amp=FALSE,
a0=0.01,b0=0.01,ci=0.95,LabelParTheta=NULL)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event. The value 1 indicates failure.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.
`amp`	the interval width is taken in account in the estimation of parameter w which controls the loss of information over time, only for the PEM. For more details see Santos et al. (2017). Default: FALSE. Optional argument.
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`ci`	the nominal level of confidence interval for the parameters. Default: ci=0.95. Optional argument.
`LabelParTheta`	If not NULL, the static parameters are called by the specified label. Optional argument.

Details

Typical usages are

```
fit=ngssm.mle(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model="Poisson",StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01),
a0=0.01,b0=0.01,ci=0.95)
```

Value

`[[1]]`	The function returns the output of the maximum likelihood estimation for the NGSSM.

Note

The function provides the MLE estimates for the static parameters of the specified model. The likelihood function is maximized using the 'optim' function and 'BFGS' method.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

FilteringF SmoothingF ngssm.bayes

Examples

```
############################################################
##
## Poisson Example: the Polio data
##
############################################################
#library(NGSSEML)
# MLE estimation:
data(Yt)
Xtm=Yt[,3:7]    # Xt as matrix always!
Xtm[,1]=(1:168-73)/168
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
#LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01)
a0=0.01
b0=0.01
ci=0.95
#Fit:
fit=ngssm.mle(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
```

```
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar,a0=a0,b0=b0,ci=ci)

##########################################################
##
## GED Example: the PETR3 data
##
##########################################################
# MLE estimation:
data(Rt)
Ytm=Rt$Rt
Xt=NULL
Zt=NULL
model="GED"
#LabelParTheta=c("w","nu")
StaPar=c(0.9,1)
a0=0.01
b0=0.01
ci=0.95
fit=ngssm.mle(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,a0=a0,b0=b0,ci=ci)

##########################################################
#
## PEM Example: the GTE data
##
##########################################################
# MLE estimation:
#library(NGSSEML)
data(gte_data)
Ytm=gte_data$V1
Xtm=NULL
Ztm=NULL
model="PEM"
amp=FALSE
Eventm=gte_data$V2        # Event: failure, 1.
Break=GridP(Ytm, Eventm, nT = NULL)
#LabelParTheta=c("w")
StaPar=c(0.73)
a0=0.01
b0=0.01
ci=0.95
fit=ngssm.mle(Ytm~1,data=data.frame(Ytm,Eventm),model=model,nBreaks=NULL,
amp=amp,a0=a0,b0=b0,ci=ci)


##########################################################
##
##  SR GAMMA MODEL: the SYS1 data
##
##########################################################
# MLE estimation:
#library(NGSSEML)
data(sys1_data)
Ytm=sys1_data[,1]+0.00001
Xtm=sys1_data[,2]   # Xt as matrix always!
Zt=NULL
model="SRWeibull"
#LabelParTheta=c("w","alpha","Beta1")
```

```
StaPar=c(0.9,0.7,0.01)
fit=ngssm.mle(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=StaPar,a0=0.01,b0=0.01,ci=0.95)
```

NumFail *Number of failures in each interval*

Description

The function NumFail provides the number of failures in each interval of the PEM.

Usage

```
NumFail(StaPar,Yt,Event,Break,Xt)
```

Arguments

StaPar	is the static parameter vector.
Yt	is a vector with the failure/censored times.
Event	is a censoring indicator (a vector).
Break	is a vector with the interval limits.
Xt	are the explanatory variables to be inserted in the model.

Details

Typical usages are

```
NumFail(StaPar,Yt,Event,Break,Xt)
```

Value

nf	'nf' is a vector with the number of failures in each interval.

Note

The function provides the number of failures in each interval.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

LikeF SmoothingF

Examples

```
################################################################################
#
## PEM Example: the GTE data
##
################################################################################
#library(NGSSEML)
data(gte_data)
Yt=gte_data$V1
Xt=NULL
Zt=NULL
amp=FALSE
Event=gte_data$V2        # Event: failure, 1.
Break=GridP(Yt, Event, nT = NULL)
StaPar=c(0.7)
NumFail(StaPar,Yt,Event, Break, Xt = NULL)
```

PlotF *Plot Function*

Description

The function PlotF provides graphs with smoothed/filtered estimates of the latent states.

Usage

```
PlotF(formula, data,na.action="na.omit",pz=NULL,nBreaks=NULL,
plotYt=TRUE,axisxdate=NULL,transf=1,model="Poisson",posts,Proc="Smooth",
Type="Marg",distl="PRED",a0=0.01,b0=0.01,ci=0.95,startdate=NULL,enddate=NULL,
Freq=NULL,Typeline='l',cols=c("black","blue","lightgrey"),xxlab="t",
yylab=expression(paste(hat(mu)[t])),xxlim=NULL,yylim=NULL,Lty=c(1,2,1),
Lwd=c(2,2,2),Cex=0.68)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event. The value 1 indicates failure.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.

`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`transf`	This argument allows the user to apply a transformation (exponentiation) in the estimates of the latent states. For example, the inverse transformation, i. e., transf = -1. The default value is 1. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`posts`	A sample or an estimate of the static parameters.
`plotYt`	If true, the time series Yt is inserted in the plot. The default value is TRUE. Optional argument.
`axisxdate`	a date vector for the x-axis can be specified in this function. The default value is NULL. Optional argument.
`Proc`	the chosen distribution of the lantent states. There are 2 options: conditional ("Cond") on the static parameters and marginal ("Marg"). The default is conditional ("Marg").
`Type`	the latent states distribution to be returned. There are 2 options: the smoothed ("Smooth") and filtering ("Filter") distributions. the chosen distribution of the lantent states. There are 2 options: conditional ("Cond") on the static parameters and marginal ("Marg"). The default is conditional ("Marg").
`distl`	the chosen distribution of the lantent states. There are 2 options: conditional on the static parameters and marginal ("Marg"). The default is conditional ("Cond").
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`ci`	the nominal level of confidence interval for the parameters. Optional argument. Default: ci=0.95.
`startdate`	If the argument axisxdate is not NULL, it is necessary to specify a start date. Optional argument.
`enddate`	If the argument axisxdate is not NULL, it is necessary to specify an end date. Optional argument.
`Freq`	If the argument axisxdateis not NULL, it is necessary to specify a frequency of the data. Optional argument.
`Typeline`	the type of plot should be drawn. Possible types are "p" for points, "l" for lines, "s" for stair steps and etc. Optional argument.
`cols`	You can specify colors in the graph. Optional argument.
`xxlab`	a title for the x-axis. Optional argument.
`yylab`	a title for the y-axis. Optional argument.
`xxlim`	a numeric vector with limits for the x-axis. Optional argument.
`yylim`	a numeric vector with limits for the y-axis. Optional argument.
`Lty`	A line type. Optional argument.

Lwd	Line width relative to the default (default=1).2 is twice as wide. Optional argument.
Cex	number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, 0.5 is 50% smaller, etc. . Optional argument.

Details

Typical usages are

```
PlotF(YYtm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model="Poisson",StaPar=estopt,axisxdate=x,Proc="Smooth",
Type="Cond",distl="FILTER",a0=0.01,b0=0.01,ci=0.95,posts=estopt,
startdate="1970/01/01",enddate="1983/12/31",Freq="months",
cols=c("black","blue","lightgrey"),xxlab="t",yylab="Yt",yylim=c(0,15),
Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68
```

Value

graph	This function returns an graph with smoothed or filtered estimates of the latent states.

Note

The model options are the Poisson, Normal, Laplace, GED, Gamma, Weibull and Generalized Gamma models. 'Zt' are the explanatory time series only for the Normal, Laplace and GED volatility models.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

FilteringF SmoothingF ngssm.bayes ngssm.mle

Examples

```
##################################################################################
## Polio data:

#library(NGSSEML)
data(Yt)
Xtm=Yt[,3:7]    # Xt as matrix always!
Xtm[,1]=(1:168-73)/168
Ytm=Yt$y
```

```
Ztm=NULL
model="Poisson"
#CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.79,-0.002,-0.11,-0.49,0.18,-0.38)
data1=data.frame(Ytm,Xtm)
#Fit:
fit=ngssm.mle(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar)
StaPar=fit[[1]][1:6]
MeanSmooth=SmoothingF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar,samples=1000,ci=0.95,
splot=FALSE)
#MeanSmooth
filpar=FilteringF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model=model,pz=NULL,StaPar=StaPar,a0=0.01,b0=0.01,splot=FALSE)

#Smoothing:
#PlotF function:
x = seq(as.Date("1970/01/01"),as.Date("1983/12/31"),"months")
PlotF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,posts=StaPar,axisxdate=x,
Proc="Smooth",Type="Cond",
distl="FILTER",a0=0.01,b0=0.01,ci=0.95,startdate="1970/01/01",
enddate="1983/12/31",Freq="months",cols=c("black","blue","lightgrey"),
xxlab="t",yylab="Yt",yylim=c(0,15),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

################################################################################

################################################################################
## Petro data:
##GED
#library(NGSSEML)
#### Inputs:
data(Rt)
Ytm=Rt$Rt
Date=Rt$Date
Xtm=NULL
Ztm=NULL
model="GED"
LabelParTheta=c("W","nu")
StaPar=c(0.9,1)
p=length(StaPar)
nn=length(Ytm)
a0=0.01
b0=0.01
pointss=30    ### points
nsamplex=1000 ## Sampling posterior
ci=0.95       # Cred. level
fitbayes=ngssm.bayes(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
StaPar=StaPar,a0=a0,b0=b0,prw=c(1,1),
prnu=c(0.01,0.01),ci=ci,pointss=pointss,nsamplex=nsamplex,
postplot=FALSE,contourplot=FALSE,LabelParTheta=LabelParTheta)
fitbayes[[1]]
posts=fitbayes[[2]]
############
#Smoothing:
```

```
############
#PlotF function:
PlotF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,plotYt=FALSE,
transf=-0.5,Proc="Smooth",Type="Marg",distl="PRED",a0=a0,b0=b0,
ci=ci,posts=posts,startdate=NULL,enddate=NULL,Freq="days",Typeline='l',
cols=c("black","blue","lightgrey"),xxlab="t",yylab=expression(paste(hat(sigma)[t])),
yylim=c(0.02,0.10),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)
windows()
#Date:

PlotF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,plotYt=FALSE,
axisxdate=Date,transf=-0.5,Proc="Smooth",xxlab="t",yylab=expression(paste(hat(sigma)[t])),
yylim=c(0.02,0.10),Type="Marg",distl="PRED",
a0=a0,b0=b0,ci=ci,posts=posts,startdate="2000-06-01",enddate="2008-01-29",
Freq="days",Typeline='l',
cols=c("black","blue","lightgrey"),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)
windows()
#Filtering:
PlotF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
plotYt=FALSE,transf=-0.5,Proc="Filter",Type="Marg",distl="PRED",a0=a0,b0=b0,ci=ci,
posts=posts,startdate=NULL,enddate=NULL,Freq="days",Typeline='l',
cols=c("black","blue","lightgrey"),xxlab="t",yylab=expression(paste(hat(sigma)[t])),
yylim=c(0.02,0.10),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)
windows()
#Date:
PlotF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
plotYt=FALSE,axisxdate=Date,transf=-0.5,Proc="Filter",Type="Marg",distl="PRED",
a0=a0,b0=b0,ci=ci,posts=posts,startdate="2000-06-01",enddate="2008-01-29",
Freq="days",Typeline='l',
cols=c("black","blue","lightgrey"),xxlab="t",yylab=expression(paste(hat(sigma)[t])),
yylim=c(0.02,0.10),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

###############################################
## Software reliability data:
## SRWeibull
# Bayesian estimation:
#library(NGSSEML)
#### Defaults values (NULL):
#### Inputs:
data(sys1_data)
Ytm=sys1_data[,1]+0.00001
Xtm=sys1_data[,2]
model="SRWeibull"
StaPar=c(0.98,0.75,0.02)
LabelParTheta=c("w","nu","Beta")
#Fit:
fitbayes=ngssm.bayes(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=StaPar,prw=c(1,1),
prnu=c(0.1,0.1),prbetamu=rep(0,1),prbetasigma=diag(100,1,1),
pointss=10,nsamplex=1000,postplot=FALSE,contourplot=FALSE,
LabelParTheta=LabelParTheta)
fitbayes[[1]]
###############################################
# Graph and Smoothing #
#### Inputs:
posts=fitbayes[[2]]
#Smoothing:
```

```
set.seed(1000)
windows()
#PlotF function:
PlotF(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,plotYt=TRUE,transf=1/4,
axisxdate=NULL,Proc="Smooth",Type="Marg",distl="PRED",a0=0.01,b0=0.01,
ci=0.95,posts=posts,Typeline='l',cols=c("black","blue","lightgrey"),
xxlab="Number of Failures",yylab=expression(paste(y[i]^(1/4))),
yylim=c(-2,10),xxlim=c(0,139),Lty=c(1,2,1),Lwd=c(1,2,1),Cex=0.68)


##################################################################################
```

`Prediction` *The h-step-ahead forecast for the observations*

Description

The function Prediction provides the h-step-ahead forecast for the observations of SR and Poisson models.

Usage

```
Prediction(formula,data,na.action="na.omit",pz=NULL,nBreaks=NULL,
model="Poisson",StaPar=NULL,a0=0.01,b0=0.01,distl="PRED",
ci=0.95,samples=500,hh=1,Xtprev=NULL,method="MLE")
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event. The value 1 indicates failure.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.

a0	the shape parameter of the initial Gamma distribution. Optional argument. Default: a0=0.01.
b0	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
distl	the latent states distribution to be returned. Optional argument.
ci	the confidence level of the confidence interval for the states. Optional argument. Default: ci=0.95.
samples	the number of samples drawn from the predictive distributions for the observations, given a point of the static parameters (StaPar). Default: samples=500.
hh	the forecast horizon. Optional argument. Default: hh=1.
Xtprev	the future values of covariates, if there are covariates in the model. Optional argument.
method	the estimation method used, MLE or Bayesian.

Details

Typical usages are

```
Prediction(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model=model,pz=NULL,StaPar=estopt,a0=a0,b0=b0,ci=ci,distl="PRED",
samples=500,hh=hh,Xtprev=Xtprev,method="MLE")
```

Value

Mean	'Mean' is the mean of the one-step-ahead forecast distribution for the observations.
Median	'Median' is the median of the one-step-ahead forecast distribution for the observations.
Perc1	'Perc1' the the percentile of ((1-ci)/2) order of the one-step-ahead forecast distribution for the observations.
Perc2	'Perc2' is the percentile of (1-(1-ci)/2) order of the one-step-ahead forecast distribution for the observations.

Note

The model options are Poisson, SRWeibull and SRGamma models, because the remaining are the volatility and piecewise exponential/PH models.

Author(s)

Thiago Rezende dos Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

LikeF SmoothingF

Examples

```
#####################################################################################
##
##  POISSON MODEL: Polio data
##
#####################################################################################
#library(NGSSEML)
#Classical:
data(Yt)
Xtprev=as.matrix(Yt[166:168,3:7])
Xtm=as.matrix(Yt[,3:7])    # Xt as matrix always!
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01)
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
#Fit:
a0=0.01
b0=0.01
ci=0.95
fit=ngssm.mle(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar,a0=a0,b0=b0,ci=ci)
hh=3
#estopt=c(0.7935051,-1.6618989,-0.1143435,-0.4872368,0.1755574,-0.3844129)
estopt=fit[[1]][1:6]
predpar=Prediction(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=estopt,a0=a0,b0=b0,ci=ci,distl="PRED",
samples=500,hh=hh,Xtprev=Xtprev,method="MLE")
predpar
predpar$Mean
predpar$Median
predpar$Perc1
predpar$Perc2
#Bayesian:
#library(NGSSEML)
#### Inputs:
data(Yt)
Xtprev=as.matrix(Yt[166:168,3:7])
Xtprev[,1]=((166:168)-73)/168
Xt=as.matrix(Yt[,3:7])    # Xt as matrix always!
Xtm[,1]=(1:168-73)/168
Ytm=Yt$y
Ztm=NULL
model="Poisson"
hh=3
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01)
pointss=5    ### points
nsamplex=1000 ## Multinomial sampling posterior
```

```
ci=0.95       ## Cred. level
#Fit:
fitbayes=ngssm.bayes(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar,
a0=a0,b0=b0,prw=c(1,1),prbetamu=rep(0,5),prbetasigma=diag(10, 5, 5),
ci=ci,pointss=pointss,nsamplex=nsamplex,postplot=FALSE,contourplot=FALSE)
posts=fitbayes[[2]]
#Prediction:
set.seed(1000)
predpar=Prediction(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=posts,
a0=a0,b0=b0,distl="PRED",ci=0.95,samples=500,hh=hh,Xtprev=Xtprev,method="Bayes")
predpar

################################################################################
##
##  SR Weibull MODEL: the SYS1 data
##
################################################################################
#require(NGSSEML)
#Classical:
data(sys1_data)
Ytm=sys1_data[,1]+0.00001
Xtm=as.matrix(sys1_data[,2])
hh=3
Xtprev=as.matrix(135+1:hh)
model="SRWeibull"
LabelParTheta=c("w","alpha","Beta1")
StaPar=c(0.9,0.7,0.01)
#Fit:
fit=ngssm.mle(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=StaPar,a0=0.01,b0=0.01,ci=0.95)
estopt=c(0.99,0.75262104,0.02342691)
estopt=fit[[1]][1:3]
predpar=Prediction(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,StaPar=estopt,pz=NULL,a0=0.01,
b0=0.01,distl="FILTER",ci=0.95,samples=500,hh=hh,
Xtprev=Xtprev,method="MLE")
#SRGamma
predpar
log(predpar)
predpar$Mean
predpar$Median
predpar$Perc1
predpar$Perc2
#library(NGSSEML)
#Bayesian:
#### Inputs:
data(sys1_data)
Yt=sys1_data[,1]+0.00001
Xt=as.matrix(sys1_data[,2])  # Xt as matrix always!
hh=3
Xtprev=as.matrix(135+1:hh)
Zt="NULL"
#model="SRWeibull"
model="SRGamma"
LabelParTheta=c("w","alpha","Beta1")
```

```
pointss=10    ### points
nsamplex=1000 ## Multinomial sampling posterior
StaPar=c(0.9,0.7,0.01)
#Fit:
fitbayes=ngssm.bayes(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=StaPar,
prw=c(1,1),prnu=c(0.1,0.1),prbetamu=rep(0,1),prbetasigma=diag(100,1,1),
pointss=pointss,nsamplex=3000,postplot=FALSE,contourplot=FALSE)
posts=fitbayes[[2]]
#Prediction:
set.seed(1000)
predpar=Prediction(Ytm~Xtm,data=data.frame(Ytm,Xtm),
model=model,pz=NULL,StaPar=posts,
a0=0.01,b0=0.01,distl="PRED",ci=0.95,samples=500,hh=hh,
Xtprev=Xtprev,method="Bayes")
predpar
```

PriorF *Prior Function*

Description

This function computes the probability density function of the specified prior for the static parameters of the model.

Usage

```
PriorF(StaPar,model="Poisson",prw=c(1,1),prnu=NULL,prchi=NULL,prmu=NULL,
prbetamu=NULL,prbetasigma=NULL)
```

Arguments

StaPar	the static parameter vector.
model	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
prw	the numeric vector of length 2, indicating the hyperparameters of the Beta prior distribution for the parameter w. Optional argument. The default value is c(1,1), which constitutes an uninformative prior for common data sets.
prnu	the numeric vector of length 2, indicating the hyperparameters of the Gamma prior distribution for the shape parameter nu. Optional argument. The default value is NULL.
prchi	the numeric vector of length 2, indicating the hyperparameters of the Gamma prior distribution for the shape parameter chi. Optional argument. The default value is NULL.
prmu	the numeric vector of length 2, indicating the mean and standard deviation for the Gaussian prior distribution for the parameter mu. Optional argument. The default value is NULL. This prior can be used in Normal, Laplace and GED time series models.
prbetamu	the numeric vector of length p, indicating mean for the Gaussian prior distribution for the parameter beta, the regression coefficients. Optional argument. The default value is NULL.

prbetasigma
: the numeric matrix of order p by p, indicating variance-covariance matrix of the Gaussian prior distribution for the parameter beta, the regression coefficients. Optional argument. The default value is NULL.

Details

Typical usages are

```
PriorF(StaPar,model="Poisson",prw=c(1,1),prnu=NULL,prchi=NULL,
prmu=NULL,prbetamu=NULL,prbetasigma=NULL)
```

Value

llik
: This function returns the probability density of the specified prior for the static parameters of the model.

Note

The model options are the Poisson, Normal, Laplace, GED, Gamma, Weibull and Generalized Gamma models. The arguments of this function depend on the static parameters in each model to be specified to the data.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

ngssm.bayes SmoothingF

Examples

```
#library(NGSSEML)
#Trend,CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.8,-0.1,0.01,0.01,0.01,0.01)
PriorF(StaPar,model="Poisson",prw=c(1,1),prnu=NULL,prchi=NULL,prmu=NULL,
prbetamu=rep(0,5),prbetasigma=diag(4, 5, 5))
```

`ProdXtChi` *ProdXtChi*

Description

ProdXtChi computes a constant to be inserted in the likelihood of the PEM with covariate (PH model).

Usage

```
ProdXtChi(StaPar,Yt,Break,Event,Xt)
```

Arguments

`StaPar`	the static parameter vector.
`Yt`	a vector with the failure/censored times.
`Break`	a vector with the interval limits.
`Event`	a censoring indicator (a vector).
`Xt`	the explanatory variables to be inserted in the model.

Details

Typical usages are

```
ProdXtChi(StaPar,Yt,Break,Event,Xt)
```

Value

`cco`	a constant.

Note

This function is ONLY to compute a constant to be inserted in the likelihood of the PEM with covariate. It is more an internal function.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

`LikeF` `SmoothingF`

Examples

```
#library(NGSSEML)
data(gte_data)
Yt=gte_data$V1
Xt=as.matrix(1:125)
model="PEM"
amp=FALSE
Event=gte_data$V2        # Event: failure, 1.
Break=GridP(Yt, Event, nT = NULL)
LabelParTheta=c("w")
StaPar=c(0.73)
ProdXtChi(StaPar,Yt,Break,Event,Xt)
```

Rt   *Returns of the asset PETR3 (Petrobras company) in the Brazilian stock market*

Description

The return data consist of 1999 daily observations in the period of 2000/01/06 to 2008/29/01.

Usage

```
data(Rt)
```

Format

A data frame with 1999 rows and 1 variable.

Details

The data irregularity due to weekends and holidays was ignored.

Source

http://http://finance.yahoo.com/

References

https://br.advfn.com/bolsa-de-valores/bovespa/petrobras-PETR3/empresa

Examples

```
data(Rt)
```

SmoothingF *Smoothing Distribution (Procedure) of the Latent States*

Description

The function SmoothingF provides an exact sample of the posterior distribution of the latent states condiotinal on the static parameters or marginal.

Usage

```
SmoothingF(formula,data,na.action="na.omit",pz=NULL,nBreaks=NULL,
model="Poisson",StaPar=NULL,Type="Cond",a0=0.01,b0=0.01,
amp=FALSE,samples=1,ci=0.95,splot=FALSE)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. The variables are: - the time series of interest Yt (first column of the data frame). the explanatory time series to be inserted in the model. - Xt must be always specified as a matrix of order n by p (after Yt). - the explanatory time series to be inserted in the mean of volatility model. Zt must be always specified as a matrix of order n by p (after Xt). - a censoring indicator of the event (a vector), only for the PEM. If the model is the PEM, put the variable Event in the secon column of tha data frame after Yt, and he explanatory time series after the variable Event. The value 1 indicates failure.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. Optional argument.
`pz`	the number of the explanatory time series to be inserted in the mean of volatility model. Default: NULL. Optional argument.
`nBreaks`	the number of breaks used to build a vector with the interval limits, only for the PEM. Optional argument.
`model`	the chosen model for the observations. The options are: Poisson, Normal, Gamma, Weibull, Generalized Gamma, Laplace, GED and PEM models.
`StaPar`	a numeric vector of initial values for the static parameters. Optional argument.
`Type`	the chosen distribution of the lantent states. There are 2 options: conditional on the static parameters and marginal ("Marg"). The default is conditional ("Cond").
`a0`	the shape parameter of the initial Gamma distribution. Optional argument. DDefault: a0=0.01.
`b0`	the scale parameter of the initial Gamma distribution. Optional argument. Default: b0=0.01.
`amp`	the interval width is taken in account in the estimation of parameter w which controls the loss of information over time, only for the PEM. For more details see Santos et al. (2017). Default: FALSE. Optional argument.

`samples` the number of samples drawn from the joint posterior distribution of the latent states, given a point of the static parameters (StaPar). Optional argument. Default: samples = 1.

`ci` the nominal level of confidence interval for the parameters. Optional argument. Default: ci=0.95.

`splot` Create a plot with the point and interval estimates of the states. Optional argument.

Details

Typical usages are

```
SmoothingF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model="Poisson",Type="Cond",a0=0.01,b0=0.01,samples=1,ci=0.95)
```

Value

`mdata` This function returns an exact sample of the join distribution of the states. If the number of samples is greater than 1, some summaries of the state samples are returned.

Note

The model options are the Poisson, Normal, Laplace, GED, Gamma, Weibull and Generalized Gamma models. 'Zt' are the explanatory time series only for the Normal, Laplace and GED volatility models.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

FilteringF LikeF ngssm.mle ngssm.bayes

Examples

```
################################################################################
##
## CLASSICAL ESTIMATION
##
################################################################################

#################################################################################
##
## Poisson Example: the Polio data
```

```
##
################################################################################
## BFGS
#library(NGSSEML)
data(Yt)
Xtm=as.matrix(Yt[,3:7])    # Xt as matrix always!
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
#Starpar=c(0.79,-0.002,-0.11,-0.49,0.18,-0.38) # initial value
StaPar=c(0.8,-0.8,0.01,0.01,0.01,0.01)
data1=data.frame(Ytm,Xtm)
fit=ngssm.mle(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model=model,pz=NULL,StaPar=StaPar)
estopt=fit[[1]][1:6]
MeanSmooth=SmoothingF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model=model,pz=NULL,StaPar=estopt,
a0=0.01,b0=0.01,samples=1000,ci=0.95,splot=FALSE)
#MeanSmooth
filpar=FilteringF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model=model,pz=NULL,StaPar=estopt,splot=FALSE)

#Smoothing:
#PlotF function:
x = seq(as.Date("1970/01/01"),as.Date("1983/12/31"),"months")
PlotF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data1,model=model,pz=NULL,Proc="Smooth",
Type="Cond",distl="FILTER",a0=0.01,b0=0.01,ci=0.95,posts=estopt,
startdate="1970/01/01",enddate="1983/12/31",Freq="months",cols=c("black","blue","lightgrey"),
xxlab="t",yylab="Yt",yylim=c(0,15),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)


################################################################################
##
## GED Example: the PETR3 data
##
################################################################################
data(Rt)
Ytm=Rt$Rt
Date=Rt$Date
Xtm=NULL
Ztm=NULL
model="GED"
labelpar=c("w","r")
fit=ngssm.mle(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL)
estopt=fit[[1]][1:2]
set.seed(1000)
MeanSmooth=SmoothingF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,StaPar=estopt,
a0=0.01,b0=0.01,samples=1000,ci=0.95,splot=FALSE)
filpar=FilteringF(Ytm~1,data=data.frame(Ytm),model=model,
pz=NULL,StaPar=estopt,a0=0.01,b0=0.01,splot=FALSE)
#Smoothing:
set.seed(1000)
fits=SmoothingF(Ytm~1,data=data.frame(Ytm),model=model,StaPar=estopt,Type="Cond",
a0=0.01,b0=0.01,ci=0.95,splot=FALSE)
```

```
#PlotF function:
#Date:
PlotF(Ytm~1,data=data.frame(Ytm),plotYt=FALSE,axisxdate=Date,transf=-0.5,model=model,
Proc="Smooth",xxlab="t",yylab=expression(paste(hat(sigma)[t])),
yylim=c(0.02,0.10),Type="Cond",distl="FILTER",
a0=0.01,b0=0.01,ci=0.95,posts=estopt,startdate="2000-06-01",enddate="2008-01-29",
Freq="days",Typeline='l',
cols=c("black","blue","lightgrey"),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

################################################################################
##
## PEM Example: the GTE data
##
################################################################################
#library(NGSSEML)
data(gte_data)
Ytm=gte_data$V1
Xtm=NULL
Ztm=NULL
model="PEM"
amp=FALSE
Eventm=gte_data$V2
Breakm=GridP(Ytm, Eventm, nT = NULL)
LabelParTheta=c("w")
StaPar=c(0.73)
a0=0.01
b0=0.01
ci=0.95
fit=ngssm.mle(Ytm~1,data=data.frame(Ytm,Eventm),model=model,StaPar=StaPar,
nBreaks=NULL,amp=amp,a0=a0,b0=b0,ci=ci,LabelParTheta=LabelParTheta)
estopt=fit[[1]][1]
set.seed(10)
MeanSmooth=SmoothingF(Ytm~1,data=data.frame(Ytm,Eventm),model=model,
a0=0.01,b0=0.1,amp=FALSE,samples=1000,ci=0.95,splot=FALSE,StaPar=estopt)

#Smoothing:
set.seed(1000)
fits=SmoothingF(Ytm~1,data=data.frame(Ytm,Eventm),model=model,Type="Cond",
a0=0.01,b0=0.1,ci=0.95,samples=1,splot=FALSE,StaPar=estopt)
#PlotF function:
PlotF(Ytm~1,data=data.frame(Ytm,Eventm),model=model,plotYt=FALSE,
axisxdate=Breakm[1:17],Proc="Smooth",Type="Cond",distl="Filter",a0=0.01,b0=0.1,
ci=0.95,posts=estopt,Typeline='s',
cols=c("black","blue","lightgrey"),xxlab="Time to Failure (Days)",yylab="Failure rate",
yylim=c(0,0.008),xxlim=c(0,139),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

################################################################################
##
## BAYESIAN ESTIMATION
##
################################################################################

################################################################################
##Poisson
## Polio Data
################################################################################
```

```
#library(NGSSEML)
#### Inputs:
data(Yt)
Xtm=as.matrix(Yt[,3:7])    # Xt as matrix always!
Ytm=Yt$y
Ztm=NULL
model="Poisson"
#CosAnnual,SinAnnual,CosSemiAnnual,SinSemiAnnual
LabelParTheta=c("w","Beta1","Beta2","Beta3","Beta4","Beta5")
StaPar=c(0.79,-0.002,-0.11,-0.49,0.18,-0.38)
a0=0.01
b0=0.01
pointss=5     ### points
nsamplex=1000 ## Multinomial sampling posterior
ci=0.95       ## Cred. level
#Fit:
#Bayesian:
fitbayes=ngssm.bayes(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=StaPar,
a0=a0,b0=b0,prw=c(1,1),prbetamu=rep(0,5),prbetasigma=diag(10, 5, 5),
ci=ci,pointss=pointss,nsamplex=nsamplex,LabelParTheta=LabelParTheta)
fitbayes[[1]]
posts=fitbayes[[2]]

#Smoothing:
set.seed(1000)
nn=length(Ytm)
fits=SmoothingF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,StaPar=posts,Type="Marg",
a0=0.01,b0=0.01,ci=0.95,splot=FALSE)
#PlotF function:
x = seq(as.Date("1970/01/01"),as.Date("1983/12/31"),"months")
PlotF(Ytm~Trend+CosAnnual+SinAnnual+CosSemiAnnual+SinSemiAnnual,
data=data.frame(Ytm,Xtm),model=model,pz=NULL,axisxdate=x,Proc="Smooth",
Type="Marg",distl="PRED",a0=a0,b0=b0,ci=ci,posts=posts,startdate="1970/01/01",
enddate="1983/12/31",Freq="months",cols=c("black","blue","lightgrey"),xxlab="t",
yylab="Yt",yylim=c(0,15),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

################################################################################
##GED
## Petro3 Data
################################################################################
#library(NGSSEML)
  #### Inputs:
  data(Rt)
  Ytm=Rt$Rt
  Date=Rt$Date
  Xtm=NULL
  Ztm=NULL
  model="GED"
  LabelParTheta=c("W","nu")
  StaPar=c(0.9,1)
  p=length(StaPar)
  nn=length(Yt)
  a0=0.01
  b0=0.01
  pointss=10    ### points
```

```
  nsamplex=1000 ## Multinomial sampling posterior
  ci=0.95        # Cred. level
  #Bayesian:
  fitbayes=ngssm.bayes(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
  StaPar=StaPar,a0=a0,b0=b0,prw=c(1,1),
  prnu=c(0.01,0.01),ci=ci,pointss=pointss,nsamplex=nsamplex,
  postplot=TRUE,contourplot=FALSE,LabelParTheta=LabelParTheta)
  posts=fitbayes[[2]]
  #Smoothing:
  set.seed(1000)
  fits=SmoothingF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
  StaPar=posts,Type="Marg",a0=a0,b0=b0,ci=ci,splot=FALSE)
  n1=length(Ytm)
  ytm=matrix(0,n1,3)

  #PlotF function:
  #Date:
  PlotF(Ytm~1,data=data.frame(Ytm),model=model,pz=NULL,
  plotYt=FALSE,axisxdate=Date,transf=-0.5,Proc="Smooth",xxlab="t",
  yylab=expression(paste(hat(sigma)[t])),
  yylim=c(0.02,0.10),Type="Marg",distl="PRED",
  a0=a0,b0=b0,ci=ci,posts=posts,startdate="2000-06-01",enddate="2008-01-29",
  Freq="days",Typeline='l',
  cols=c("black","blue","lightgrey"),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

################################################################################
##PEM
##GTE Data
################################################################################
#library(NGSSEML)
#### Inputs:
data(gte_data)
Ytm=gte_data$V1
Eventm=gte_data$V2
Breakm=GridP(Ytm, Eventm, nT = NULL)
Xtm=NULL
Ztm=NULL
model="PEM"
amp=FALSE
LabelParTheta=c("w")
StaPar=c(0.73)
p=length(StaPar)
nn=length(Breakm)
a0=0.01
b0=0.1
p=length(StaPar)
linf=numeric(p)
lsup=numeric(p)
pointss=30    ### points
nsamplex=1000 ## Multinomial sampling posterior
ci=0.95
alpha=1-ci
#Fit:
#Bayesian:
fitbayes=ngssm.bayes(Ytm~1,data=data.frame(Ytm,Eventm),model=model,
pz=NULL,StaPar=StaPar,amp=amp,a0=a0,b0=b0,prw=c(1,1),prnu=NULL,prchi=NULL,
prmu=NULL,prbetamu=NULL,prbetasigma=NULL,ci=ci,pointss=pointss,nsamplex=nsamplex,
```

```
postplot=FALSE,contourplot=FALSE,LabelParTheta=LabelParTheta)
fitbayes[[1]]
posts=fitbayes[[2]]

#Smoothing:
set.seed(1000)
fits=SmoothingF(Ytm~1,data=data.frame(Ytm,Eventm),model=model,pz=NULL,
StaPar=posts,Type="Marg",a0=a0,b0=b0,ci=ci,samples=1,splot=FALSE)
#PlotF function:
PlotF(Ytm~1,data=data.frame(Ytm,Eventm),model=model,pz=NULL,plotYt=FALSE,
axisxdate=Breakm[1:17],Proc="Smooth",Type="Marg",distl="PRED",a0=a0,b0=b0,
ci=ci,posts=posts,Typeline='s',
cols=c("black","blue","lightgrey"),xxlab="Time to Failure (Days)",yylab="Failure rate",
yylim=c(0,0.008),xxlim=c(0,139),Lty=c(1,2,1),Lwd=c(2,2,2),Cex=0.68)

################################################################################
```

---

sys1_data *The times between successive computer software failures of the SYS1*

Description

The times between 136 successive computer software failures and the number of failures of the SYS1 data.

Usage

```
data(sys1_data)
```

Format

A data frame with 136 rows and 2 variables.

Details

The first column of the object sys1_data corresponds to the times and the second to the number of detected failures before the i-th stage.

Source

Lyu, M. R. (1996). Handbook of software reliability engineering.

References

Lyu, M. R. (1996). Handbook of software reliability engineering.

Examples

```
data(sys1_data)
```

`TTime` *Total Failure Time*

Description

The function TTime provides a vector with the contribution of the units for each interval of the PEM.

Usage

```
TTime(StaPar,Yt,Event, Break, Xt = NULL)
```

Arguments

`StaPar`	the static parameter vector.
`Yt`	a vector with the failure/censored times.
`Event`	a censoring indicator (a vector).
`Break`	a vector with the interval limits.
`Xt`	the explanatory variables to be inserted in the model. Optional argument.

Details

Typical usages are

```
TTime(StaPar,Yt,Event, Break, Xt = NULL)
```

Value

`ttime`	a vector with the contribution of the units for each interval.

Note

This function provides the total failure time.

Author(s)

T. R. Santos

References

Gamerman, D., Santos, T. R., and Franco, G. C. (2013). A Non-Gaussian Family of State-Space Models with Exact Marginal Likelihood. Journal of Time Series Analysis, 34(6), 625-645.

Santos T. R., Gamerman, D., Franco, G. C. (2017). Reliability Analysis via Non-Gaussian State-Space Models. IEEE Transactions on Reliability, 66, 309-318.

See Also

`LikeF` `SmoothingF`

Examples

```
################################################################################
#
## PEM Example: the GTE data
##
################################################################################
#library(NGSSEML)
data(gte_data)
Yt=gte_data$V1
Xt=NULL
Zt=NULL
amp=FALSE
Event=gte_data$V2        # Event: failure, 1.
Break=GridP(Yt, Event, nT = NULL)
StaPar=c(0.7)
TTime(StaPar,Yt,Event, Break, Xt = NULL)
```

`Yt` *The Polio Data*

Description

The data consist of monthly counts of poliomyelitis cases in the USA from the year 1970 to 1983.

Usage

```
data(Yt)
```

Format

A data frame with 168 observations on the following 8 variables.

Details

The covariates are the deterministic trend centered at 73 and divided by 1000, annual and semiannual cosine and annual and semiannual sine.

Source

Centers for Disease Control, USA.

References

Zeger, S.L. (1988). A regression model for time series of counts. Biometrika 75, 621-29.

Examples

```
data(Yt)
```

Index