Package 'exdqlm'

January 12, 2026

Title Extended Dynamic Quantile Linear Models

Version 0.2.0

Author Raquel Barata [aut, cre],
Raquel Prado [ths],
Bruno Sanso [ths],
Antonio Aguirre [aut]

Maintainer Raquel Barata <raquel.a.barata@gmail.com>

Description Routines for Bayesian estimation and analysis of dynamic quantile linear models utilizing the extended asymmetric Laplace error distribution, also known as extended dynamic quantile linear models (exDQLM) described in Barata et al (2020) <doi:10.1214/21-AOAS1497>.

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 2.10)

Imports stats, methods, graphics, coda, tictoc, magic, crch,
truncnorm, HyperbolicDist, GeneralizedHyperbolic, FNN,
LaplacesDemon, grDevices, Rcpp, matrixStats

Repository CRAN

Date/Publication 2026-01-12 19:00:08 UTC

Suggests rmarkdown, testthat (>= 3.0.0), ggplot2, pkgload

Config/testthat/edition 3

LinkingTo BH, Rcpp, RcppArmadillo

NeedsCompilation yes

Contents

`+.exdqlm` *Addition for* `exdqlm` *objects*

Description

Combines two state space blocks into a single state space model for an exDQLM.

Usage

```
## S3 method for class 'exdqlm'
m1 + m2
```

Arguments

m1	object of class "exdqlm" containing the first model to be combined.
m2	object of class "exdqlm" containing the second model to be combined.

Value

A object of class "exdqlm" containing the new combined state space model components:

- FF - Observational vector.
- GG - Evolution matrix.
- m0 - Prior mean of the state vector.
- C0 - Prior covariance of the state vector.

Examples

```
trend.comp = polytrendMod(2,rep(0,2),10*diag(2))
seas.comp = seasMod(365,c(1,2,4),C0=10*diag(6))
model = trend.comp + seas.comp
```

---

as.exdqlm exdqlm *objects*

Description

as.exdqlm attempts to turn a list into an exdqlm object. Works for time-invariant dlm objects created using the **dlm** package.

Usage

```
as.exdqlm(m)
```

Arguments

m	a list containing named elements m0, C0, FF and GG.

Value

A object of class "exdqlm" containing the state space model components:

- FF - Observational vector.
- GG - Evolution matrix.
- m0 - Prior mean of the state vector.
- C0 - Prior covariance of the state vector.

BTflow — *Monthly time-series of water flow at Big Tree water gauge.*

Description

Average monthly natural water flow (cubic feet per second) at the Big Tree gauge of the San Lorenzo river in Santa Cruz County, CA from 1937 through 2014.

Usage

```
BTflow
```

Format

A time series of length 936.

Source

https://waterdata.usgs.gov/nwis/

References

U.S. Geological Survey (2016). National Water Information System data available on the World Wide Web (USGS Water Data for the Nation). https://waterdata.usgs.gov/nwis/.

compPlot — *Plot a component of an exDQLM*

Description

The function plots the dynamic MAP estimates and 95% credible intervals (CrIs) of a specified component of an exDQLM. Alternatively, if `just.theta=TRUE` the MAP estimates and 95% credible intervals (CrIs) of a single element of the dynamic state vector are plotted.

Usage

```
compPlot(
  m1,
  index,
  add = FALSE,
  col = "purple",
  just.theta = FALSE,
  cr.percent = 0.95
)
```

Arguments

m1	An object of class "exdqlmMCMC" or "exdqlmISVB".
index	Index of the component or element of the state vector to be plotted.
add	If TRUE, the dynamic component will be added to existing plot.
col	Color of dynamic component to be plotted. Default is purple.
just.theta	If TRUE, the function plots the dynamic distribution of the index element of the state vector. If just.theta=TRUE, index must have length 1.
cr.percent	Percentage used in the calculation of the credible intervals.

Value

A list of the following is returned:

- map.comp - MAP estimate of the dynamic component (or element of the state vector).
- lb.comp - Lower bound of the 95% CrIs of the dynamic component (or element of the state vector).
- ub.comp - Upper bound of the 95% CrIs of the dynamic component (or element of the state vector).

Examples

```
y = scIVTmag[1:365]
trend.comp = polytrendMod(2,rep(0,2),10*diag(2))
seas.comp = seasMod(365,c(1,2,4),C0=10*diag(6))
model = trend.comp + seas.comp
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.98,1),dim.df = c(2,6),
                  gam.init=-3.5,sig.init=15,tol=0.05)
# plot first harmonic component
compPlot(M0,index=c(3,4),col="blue")
```

---

dexal *Density Function for the Extended Asymmetric Laplace (exAL) Distribution*

Description

Computes the PDF of the Extended Asymmetric Laplace (exAL) distribution. Vectorized over x.

Usage

```
dexal(x, p0 = 0.5, mu = 0, sigma = 1, gamma = 0, log = FALSE)
```

Arguments

x	Numeric vector of quantiles.
p0	Probability level used in the quantile parametrization. Scalar in (0, 1). Default 0.5.
mu	Location parameter (scalar). Default 0.
sigma	Scale parameter (scalar, strictly positive). Default 1.
gamma	Skewness parameter controlling asymmetry (scalar). Must be within valid bounds implied by p0. Default 0.
log	Logical scalar; if TRUE return log-density. Default FALSE.

Value

Numeric vector of densities (same length as x).

Examples

```
dexal(0)
dexal(1, p0 = 0.75, mu = 0, sigma = 2, gamma = 0.25)
dexal(seq(-3, 3, by = 0.1), p0 = 0.3, mu = 0, sigma = 1, gamma = -0.45)
```

ELIanoms *Daily time-series of ELI anomalies.*

Description

ELI anomalies on the daily time-scale from January 1, 1979 to December 31, 2019 with all February 29ths omitted.

Usage

```
ELIanoms
```

Format

A time series of length 14965.

Source

https://portal.nersc.gov/archive/home/projects/cascade/www/ELI

References

Patricola, C.M., O'Brien, J.P., Risser, M.D. et al. *Maximizing ENSO as a source of western US hydroclimate predictability.* Clim Dyn 54, 351–372 (2020). doi:10.1007/s00382019050048

`exdqlmDiagnostics` *exDQLM Diagnostics*

Description

The function computes the following for the model(s) provided: the posterior predictive loss criterion based off the check loss, the one-step-ahead distribution sequence and its KL divergence from normality. The function also plots the following: the qq-plot and ACF plot corresponding to the one-step-ahead distribution sequence, and a time series plot of the MAP standard forecast errors.

Usage

```
exdqlmDiagnostics(
  m1,
  m2 = NULL,
  plot = TRUE,
  cols = c("red", "blue"),
  ref = NULL
)
```

Arguments

`m1`	An object of class "`exdqlmMCMC`" or "`exdqlmISVB`".
`m2`	An optional additional object of class "`exdqlmMCMC`" or "`exdqlmISVB`" to compare with `m1`.
`plot`	If `TRUE`, the following will be plotted for `m1` and `m2` (if provided): a qq-plot and ACF plot of the MAP one-step-ahead distribution sequence, and a time series plot of the standardized forecast errors.
`cols`	Color(s) used to plot diagnostics.
`ref`	Reference sample of size `length(y)` from a standard normal distribution used to compute the KL divergence.

Value

A object of class "`exdqlmDiagnostics`" containing the following:

- `m1.uts` - The one-step-ahead distribution sequence of `m1`.
- `m1.KL` - The KL divergence of `m1.uts` and a standard normal.
- `m1.pplc` - The posterior predictive loss criterion of `m1` based off the check loss function.
- `m1.qq` - The ordered pairs of the qq-plot comparing `m1.uts` with a standard normal distribution.
- `m1.acf` - The autocorrelations of `m1.uts` by lag.
- `m1.rt` - Run-time of the original model `m1` in seconds.
- `m1.msfe` - MAP standardized one-step-ahead forecast errors from the original model `m1`.
- `y` - The original time-series used to fit `m1`.

If `m2` is provided, analogous results for `m2` are also included in the list.

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,mean(y),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.95),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
M0.diags = exdqlmDiagnostics(M0,plot=FALSE)
```

`exdqlmForecast` *k-step-ahead quantile forecasts*

Description

Computes filtered and k-step-ahead forecast quantiles from a fitted dynamic quantile model and optionally adds them to an existing plot.

Usage

```
exdqlmForecast(
  start.t,
  k,
  m1,
  fFF = NULL,
  fGG = NULL,
  plot = TRUE,
  add = FALSE,
  cols = c("purple", "magenta"),
  cr.percent = 0.95
)
```

Arguments

`start.t`	Integer index at which forecasts start (must be within the span of the fitted model in `m1`).
`k`	Integer; number of steps ahead to forecast.
`m1`	A fitted exDQLM model object, typically returned by `exdqlmISVB()` or `exdqlmMCMC()`.
`fFF`	Optional state vector(s) for the forecast steps. A numeric matrix with p rows and either 1 column (non–time-varying) or k columns (time-varying). Its dimension must match the fitted model in `m1`.
`fGG`	Optional evolution matrix/matrices for the forecast steps. Either a numeric $p \times p$ matrix (non–time-varying) or a $p \times p \times k$ array (time-varying). Its dimensions must match the fitted model in `m1`.
`plot`	Logical; if `TRUE`, plot filtered and forecast quantiles with equal–tailed credible intervals. Default `TRUE`.

add	Logical; if TRUE, add the forecasted quantiles to the current plot. Default FALSE.
cols	Character vector of length 2 giving the colors for filtered and forecasted quantiles respectively. Default c("purple","magenta").
cr.percent	Numeric in (0, 1); the probability mass for the credible intervals (e.g., 0.95). Default 0.95.

Value

A object of class "exdqlmForecast" containing the following:

- start.t Integer index at which forecasts start (within the span of the fitted model in m1).
- k Integer; number of steps ahead forecasted.
- m1 A fitted exDQLM model object, typically returned by exdqlmISVB() or exdqlmMCMC().
- cr.percent Numeric in (0, 1); the probability mass for the credible intervals (e.g., 0.95).
- fa Forecast state mean vectors ($p \times k$ matrix).
- fR Forecast state covariance matrices ($p \times p \times k$ array).
- ff Forecast quantile means (length-k numeric).
- fQ Forecast quantile variances (length-k numeric).

Examples

```
# Toy example; keep small and fast
y <- scIVTmag[1:100]
model <- polytrendMod(1, stats::quantile(y, 0.85), 10)
M0 <- exdqlmISVB(y, p0 = 0.85, model, df = c(0.98), dim.df = c(1),
                 gam.init = -3.5, sig.init = 15)
exdqlmForecast(start.t = 90, k = 10, m1 = M0)
```

---

exdqlmISVB *exDQLM - ISVB algorithm*

Description

The function applies an Importance Sampling Variational Bayes (ISVB) algorithm to estimate the posterior of an exDQLM.

Usage

```
exdqlmISVB(
  y,
  p0,
  model,
  df,
  dim.df,
```

```
  fix.gamma = FALSE,
  gam.init = NA,
  fix.sigma = TRUE,
  sig.init = NA,
  dqlm.ind = FALSE,
  exps0,
  tol = 0.1,
  n.IS = 500,
  n.samp = 200,
  PriorSigma = NULL,
  PriorGamma = NULL,
  verbose = TRUE
)
```

Arguments

y	A univariate time-series.
p0	The quantile of interest, a value between 0 and 1.
model	List of the state-space model including GG, FF, prior parameters m0 and C0.
df	Discount factors for each block.
dim.df	Dimension of each block of discount factors.
fix.gamma	Logical value indicating whether to fix gamma at gam.init. Default is FALSE.
gam.init	Initial value for gamma (skewness parameter), or value at which gamma will be fixed if fix.gamma=TRUE.
fix.sigma	Logical value indicating whether to fix sigma at sig.init. Default is TRUE.
sig.init	Initial value for sigma (scale parameter), or value at which sigma will be fixed if fix.sigma=TRUE.
dqlm.ind	Logical value indicating whether to fix gamma at 0, reducing the exDQLM to the special case of the DQLM. Default is FALSE.
exps0	Initial value for dynamic quantile. If exps0 is not specified, it is set to the DLM estimate of the p0 quantile.
tol	Tolerance for convergence of dynamic quantile estimates. Default is tol=0.1.
n.IS	Number of particles for the importance sampling of joint variational distribution of sigma and gamma. Default is n.IS=500.
n.samp	Number of samples to draw from the approximated posterior distribution. Default is n.samp=200.
PriorSigma	List of parameters for inverse gamma prior on sigma; shape a_sig and scale b_sig. Default is an inverse gamma with mean 1 (or sig.init if provided) and variance 10.
PriorGamma	List of parameters for truncated student-t prior on gamma; center m_gam, scale s_gam and degrees of freedom df_gam. Default is a standard student-t with 1 degree of freedom, truncated to the support of gamma.
verbose	Logical value indicating whether progress should be displayed.

Value

A object of class "exdqlmISVB" containing the following:

- y - Time-series data used to fit the model.
- run.time - Algorithm run time in seconds.
- iter - Number of iterations until convergence was reached.
- dqlm.ind - Logical value indicating whether gamma was fixed at 0, reducing the exDQLM to the special case of the DQLM.
- model - List of the state-space model including GG, FF, prior parameters m0 and C0.
- p0 - The quantile which was estimated.
- df - Discount factors used for each block.
- dim.df - Dimension used for each block of discount factors.
- sig.init - Initial value for sigma, or value at which sigma was fixed if fix.sigma=TRUE.
- seq.sigma - Sequence of sigma estimated by the algorithm until convergence.
- samp.theta - Posterior sample of the state vector variational distribution.
- samp.post.pred - Sample of the posterior predictive distributions.
- map.standard.forecast.errors - MAP standardized one-step-ahead forecast errors.
- samp.sigma - Posterior sample of scale parameter sigma variational distribution.
- samp.vts - Posterior sample of latent parameters, v_t, variational distributions.
- theta.out - List containing the variational distribution of the state vector including filtered distribution parameters (fm and fC) and smoothed distribution parameters (sm and sC).
- vts.out - List containing the variational distributions of latent parameters v_t.
- fix.sigma Logical value indicating whether sigma was fixed at sig.init.

If dqlm.ind=FALSE, the object also contains:

- gam.init - Initial value for gamma, or value at which gamma was fixed if fix.gamma=TRUE.
- seq.gamma - Sequence of gamma estimated by the algorithm until convergence.
- samp.gamma - Posterior sample of skewness parameter gamma variational distribution.
- samp.sts - Posterior sample of latent parameters, s_t, variational distributions.
- gammasig.out - List containing the IS estimate of the variational distribution of sigma and gamma.
- sts.out - List containing the variational distributions of latent parameters s_t.
- fix.gamma Logical value indicating whether gamma was fixed at gam.init.

Or if dqlm.ind=TRUE, the object also contains:

- sig.out - List containing the IS estimate of the variational distribution of sigma.

Examples

```
y = scIVTmag[1:1095]
trend.comp = polytrendMod(1,mean(y),10)
seas.comp = seasMod(365,c(1,2,4),C0=10*diag(6))
model = trend.comp + seas.comp
M0 = exdqlmISVB(y,p0=0.85,model,df=c(1,1),dim.df = c(1,6),
                gam.init=-3.5,sig.init=15,tol=0.05)
```

`exdqlmMCMC` *exDQLM - MCMC algorithm*

Description

The function applies a Markov chain Monte Carlo (MCMC) algorithm to sample the posterior of an exDQLM.

Usage

```
exdqlmMCMC(
  y,
  p0,
  model,
  df,
  dim.df,
  fix.gamma = FALSE,
  gam.init = NA,
  fix.sigma = FALSE,
  sig.init = NA,
  dqlm.ind = FALSE,
  Sig.mh,
  joint.sample = FALSE,
  n.burn = 2000,
  n.mcmc = 1500,
  init.from.isvb = TRUE,
  PriorSigma = NULL,
  PriorGamma = NULL,
  verbose = TRUE
)
```

Arguments

`y`	A univariate time-series.
`p0`	The quantile of interest, a value between 0 and 1.
`model`	List of the state-space model including `GG`, `FF`, prior parameters `m0` and `C0`.

df	Discount factors for each block.
dim.df	Dimension of each block of discount factors.
fix.gamma	Logical value indicating whether to fix gamma at gam.init. Default is FALSE.
gam.init	Initial value for gamma (skewness parameter), or value at which gamma will be fixed if fix.gamma=TRUE.
fix.sigma	Logical value indicating whether to fix sigma at sig.init. Default is TRUE.
sig.init	Initial value for sigma (scale parameter), or value at which sigma will be fixed if fix.sigma=TRUE.
dqlm.ind	Logical value indicating whether to fix gamma at 0, reducing the exDQLM to the special case of the DQLM. Default is FALSE.
Sig.mh	Covariance matrix used in the random walk MH step to jointly sample sigma and gamma.
joint.sample	Logical value indicating whether or not to recompute Sig.mh based off the initial burn-in samples of gamma and sigma. Default is FALSE.
n.burn	Number of MCMC iterations to burn. Default is n.burn = 2000.
n.mcmc	Number of MCMC iterations to sample. Default is n.mcmc = 1500.
init.from.isvb	Logical value indicating whether or not to initialize the MCMC using the ISVB algorithm. Default is TRUE.
PriorSigma	List of parameters for inverse gamma prior on sigma; shape a_sig and scale b_sig. Default is an inverse gamma with mean 1 (or sig.init if provided) and variance 10.
PriorGamma	List of parameters for truncated student-t prior on gamma; center m_gam, scale s_gam and degrees of freedom df_gam. Default is a standard student-t with 1 degree of freedom, truncated to the support of gamma.
verbose	Logical value indicating whether progress should be displayed.

Value

A object of class "exdqlmMCMC" containing the following:

- y - Time-series data used to fit the model.
- run.time - Algorithm run time in seconds.
- model - List of the state-space model including GG, FF, prior parameters m0 and C0.
- p0 - The quantile which was estimated.
- df - Discount factors used for each block.
- dim.df - Dimension used for each block of discount factors.
- samp.theta - Posterior sample of the state vector.
- samp.post.pred - Sample of the posterior predictive distributions.
- map.standard.forecast.errors - MAP standardized one-step-ahead forecast errors.
- samp.sigma - Posterior sample of scale parameter sigma.
- samp.vts - Posterior sample of latent parameters, v_t.

- theta.out - List containing the distributions of the state vector including filtered distribution parameters (fm and fC) and smoothed distribution parameters (sm and sC).
- n.burn Number of MCMC iterations that were burned.
- n.mcmc Number of MCMC iterations that were sampled.

If dqlm.ind=FALSE, the object also contains the following:

- samp.gamma - Posterior sample of skewness parameter gamma.
- samp.sts - Posterior sample of latent parameters, s_t.
- init.log.sigma - Burned samples of log sigma from the random walk MH joint sampling of sigma and gamma.
- init.logit.gamma - Burned samples of logit gamma from the random walk MH joint sampling of sigma and gamma.
- accept.rate - Acceptance rate of the MH step.
- Sig.mh - Covariance matrix used in MH step to jointly sample sigma and gamma.

Examples

```
y = scIVTmag[1:100]
trend.comp = polytrendMod(1,mean(y),10)
seas.comp = seasMod(365,c(1,2,4),C0=10*diag(6))
model = trend.comp + seas.comp
M2 = exdqlmMCMC(y,p0=0.85,model,df=c(1,1),dim.df = c(1,6),
                gam.init=-3.5,sig.init=15,
                n.burn=100,n.mcmc=150)
```

---

exdqlmPlot *Plot exDQLM*

Description

The function plots the MAP estimates and 95% credible intervals (CrIs) of the dynamic quantile of an exDQLM.

Usage

```
exdqlmPlot(m1, add = FALSE, col = "purple", cr.percent = 0.95)
```

Arguments

m1	An object of class "exdqlmMCMC" or "exdqlmISVB".
add	If TRUE, the dynamic quantile will be added to existing plot.
col	Color of dynamic quantile to be plotted. Default is purple.
cr.percent	Percentage used in the calculation of the credible intervals.

Value

A list of the following is returned:

- `map.quant` - MAP estimate of the dynamic quantile.
- `lb.quant` - Lower bound of the 95% CrIs of the dynamic quantile.
- `ub.quant` - Upper bound of the 95% CrIs of the dynamic quantile.

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
exdqlmPlot(M0,col="blue")
```

`is.exdqlm` `exdqlm` *objects*

Description

`is.exdqlm` tests if its argument is a `exdqlm` object.

Usage

```
is.exdqlm(m)
```

Arguments

`m`	an **R** object

`is.exdqlmDiagnostic` `exdqlmDiagnostic` *objects*

Description

`is.exdqlmDiagnostic` tests if its argument is a `exdqlmDiagnostic` object.

Usage

```
is.exdqlmDiagnostic(x)
```

Arguments

`x`	an **R** object

is.exdqlmForecast exdqlmForecast *objects*

Description

is.exdqlmForecast tests if its argument is a exdqlmForecast object.

Usage

```
is.exdqlmForecast(x)
```

Arguments

x	an **R** object

is.exdqlmISVB exdqlmISVB *objects*

Description

is.exdqlmISVB tests if its argument is a exdqlmISVB object.

Usage

```
is.exdqlmISVB(m)
```

Arguments

m	an **R** object

is.exdqlmMCMC exdqlmMCMC *objects*

Description

is.exdqlmMCMC tests if its argument is a exdqlmMCMC object.

Usage

```
is.exdqlmMCMC(m)
```

Arguments

m	an **R** object

nino34 *Monthly Niño 3.4 Index.*

Description

Monthly Niño 3.4 sea surface temperature (SST) Index time series.

Usage

```
nino34
```

Format

A time series of length 936.

Source

https://psl.noaa.gov/gcos_wgsp/Timeseries/Nino34/

References

Rayner N. A., D. E. Parker, E. B. Horton, C. K. Folland, L. V. Alexander, D. P. Rowell, E. C. Kent, A. Kaplan. *Global analyses of sea surface temperature, sea ice, and night marine air temperature since the late nineteenth century*. J. Geophys. Res., 108 (D14), 4407 (2003). doi:10.1029/2002JD002670

pexal *Cumulative Distribution Function (CDF) for the exAL Distribution*

Description

Vectorized over q.

Usage

```
pexal(
  q,
  p0 = 0.5,
  mu = 0,
  sigma = 1,
  gamma = 0,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

q	Numeric vector of quantiles.
p0	Probability level used in the quantile parametrization. Scalar in (0, 1). Default 0.5.
mu	Location parameter (scalar). Default 0.
sigma	Scale parameter (scalar, strictly positive). Default 1.
gamma	Skewness parameter controlling asymmetry (scalar). Must be within valid bounds implied by p0. Default 0.
lower.tail	Logical scalar; if TRUE (default) return $P(X \leq q)$, otherwise $P(X > q)$.
log.p	Logical scalar; if TRUE, return log-probabilities.

Value

Numeric vector of CDF values (same length as q).

Examples

```
pexal(0)
pexal(c(-1, 0, 1), p0 = 0.5, mu = 0, sigma = 1, gamma = 0.1)
```

---

plot.exdqlmDiagnostic *Plot Method for* exdqlmDiagnostic *Objects*

Description

Plot Method for exdqlmDiagnostic Objects

Usage

```
## S3 method for class 'exdqlmDiagnostic'
plot(x, ...)
```

Arguments

x	An exdqlmDiagnostic object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,mean(y),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.95),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
M0.diags = exdqlmDiagnostics(M0,plot=FALSE)
plot(M0.diags)
```

plot.exdqlmForecast *Plot Method for* exdqlmForecast *Objects*

Description

Plot Method for exdqlmForecast Objects

Usage

```
## S3 method for class 'exdqlmForecast'
plot(x, ...)
```

Arguments

x	An exdqlmForecast object.
...	Additional arguments (unused).

Examples

```
y <- scIVTmag[1:100]
model <- polytrendMod(1, stats::quantile(y, 0.85), 10)
M0 <- exdqlmISVB(y, p0 = 0.85, model, df = c(0.98), dim.df = c(1),
                 gam.init = -3.5, sig.init = 15)
M0.forecast = exdqlmForecast(start.t = 90, k = 10, m1 = M0)
plot(M0.forecast)
```

plot.exdqlmISVB *Plot Method for* exdqlmISVB *Objects*

Description

Plot Method for exdqlmISVB Objects

Usage

```
## S3 method for class 'exdqlmISVB'
plot(x, ...)
```

Arguments

x	An exdqlmISVB object.
...	Additional arguments.

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
plot(M0)
```

`plot.exdqlmMCMC` *Plot Method for* `exdqlmMCMC` *Objects*

Description

Plot Method for `exdqlmMCMC` Objects

Usage

```
## S3 method for class 'exdqlmMCMC'
plot(x, ...)
```

Arguments

`x`	An `exdqlmMCMC` object.
`...`	Additional arguments.

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M2 = exdqlmMCMC(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                gam.init=-3.5,sig.init=15,
                n.burn=100,n.mcmc=150)
plot(M2)
```

`polytrendMod` *Create an n-th order polynomial exDQLM component*

Description

The function creates an n-th order polynomial exDQLM component.

Usage

```
polytrendMod(order, m0, C0)
```

Arguments

order	The order of the polynomial model.
m0	Prior mean of the state vector. Default is m0 = rep(0,order).
C0	Prior covariance of the state vector. Default is C0 = 1e3*diag(order).

Value

A object of class "exdqlm" containing the following:

- FF - Observational vector.
- GG - Evolution matrix.
- m0 - Prior mean of the state vector.
- C0 - Prior covariance of the state vector.

Examples

```
# create a second order polynomial component
trend.comp = polytrendMod(2,rep(0,2),10*diag(2))
```

---

print.exdqlm *Print exDQLM model details*

Description

Print the details of the exDQLM model.

Usage

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

Arguments

x	a exdqlm object.
...	further arguments (unused).

print.exdqlmDiagnostic *Print Method for* exdqlmDiagnostic *Objects*

Description

Print Method for exdqlmDiagnostic Objects

Usage

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

Arguments

x	An exdqlmDiagnostic object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,mean(y),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.95),dim.df = c(1),
                 gam.init=-3.5,sig.init=15)
M0.diags = exdqlmDiagnostics(M0,plot=FALSE)
print(M0.diags)
```

print.exdqlmForecast *Print Method for* exdqlmForecast *Objects*

Description

Print Method for exdqlmForecast Objects

Usage

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

Arguments

x	An exdqlmForecast object.
...	Additional arguments (unused).

Examples

```
y <- scIVTmag[1:100]
model <- polytrendMod(1, stats::quantile(y, 0.85), 10)
M0 <- exdqlmISVB(y, p0 = 0.85, model, df = c(0.98), dim.df = c(1),
                 gam.init = -3.5, sig.init = 15)
M0.forecast = exdqlmForecast(start.t = 90, k = 10, m1 = M0)
print(M0.forecast)
```

---

print.exdqlmISVB *Print Method for* exdqlmISVB *Objects*

Description

Print Method for exdqlmISVB Objects

Usage

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

Arguments

x	An exdqlmISVB object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
print(M0)
```

---

print.exdqlmMCMC *Print Method for* exdqlmMCMC *Objects*

Description

Print Method for exdqlmMCMC Objects

Usage

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

Arguments

x	An exdqlmMCMC object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M2 = exdqlmMCMC(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                gam.init=-3.5,sig.init=15,
                n.burn=100,n.mcmc=150)
print(M2)
```

---

qexal *Quantile Function for the exAL Distribution*

Description

Vectorized over p.

Usage

```
qexal(
  p,
  p0 = 0.5,
  mu = 0,
  sigma = 1,
  gamma = 0,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

p	Numeric vector of probabilities in (0, 1).
p0	Probability level used in the quantile parametrization. Scalar in (0, 1). Default 0.5.
mu	Location parameter (scalar). Default 0.
sigma	Scale parameter (scalar, strictly positive). Default 1.

gamma	Skewness parameter controlling asymmetry (scalar). Must be within valid bounds implied by p0. Default 0.
lower.tail	Logical scalar; if TRUE (default) return $P(X \leq q)$, otherwise $P(X > q)$.
log.p	Logical scalar; if TRUE, return log-probabilities.

Value

Numeric vector of quantiles (same length as p).

Examples

```
p <- seq(0.1, 0.9, by = 0.2)
q <- qexal(p, p0 = 0.5, mu = 0, sigma = 1, gamma = 0)
all.equal(p, pexal(q, p0 = 0.5, mu = 0, sigma = 1, gamma = 0), tol = 1e-4)
```

---

rexal *Random Sample Generation for the exAL Distribution*

Description

Random Sample Generation for the exAL Distribution

Usage

```
rexal(n, p0 = 0.5, mu = 0, sigma = 1, gamma = 0)
```

Arguments

n	Positive integer number of samples to draw (scalar).
p0	Probability level used in the quantile parametrization. Scalar in (0, 1). Default 0.5.
mu	Location parameter (scalar). Default 0.
sigma	Scale parameter (scalar, strictly positive). Default 1.
gamma	Skewness parameter controlling asymmetry (scalar). Must be within valid bounds implied by p0. Default 0.

Value

Numeric vector of length n.

Examples

```
set.seed(1); length(rexal(10))
rexal(3, p0 = 0.5, mu = c(-1, 0, 1))
```

scIVTmag *Time series of daily average magnitude IVT in Santa Cruz, CA.*

Description

ECMWF Re-Analysis 5 (ERA5) daily average magnitude IVT in Santa Cruz, CA (approximately 22 N, 122 W) from January 1, 1979 to December 31, 2019 with all February 29ths omitted.

Usage

```
scIVTmag
```

Format

A time series of length 14965.

Source

https://cds.climate.copernicus.eu

References

Hersbach, H, Bell, B, Berrisford, P, et al. *The ERA5 global reanalysis.* Q J R Meteorol Soc. 2020; 146: 1999– 2049. doi:10.1002/qj.3803

seasMod *Create Fourier representation of a periodic exDQLM component*

Description

The function creates a Fourier form periodic component for given period and harmonics.

Usage

```
seasMod(p, h, m0, C0)
```

Arguments

p	The period.
h	Vector of harmonics to be included.
m0	Prior mean of the state vector.
C0	Prior covariance of the state vector.

Value

A object of class "exdqlm" containing the following:

- FF - Observational vector.
- GG - Evolution matrix.
- m0 - Prior mean of the state vector.
- C0 - Prior covariance of the state vector.

Examples

```
# create a seasonal component with first, second and fourth harmonics of a period of 365
seas.comp = seasMod(365,c(1,2,4),C0=10*diag(6))
```

---

summary.exdqlm *Summary exDQLM model details*

Description

Print the details of the exDQLM model.

Usage

```
## S3 method for class 'exdqlm'
summary(object, ...)
```

Arguments

object	a exdqlm object.
...	further arguments (unused).

---

summary.exdqlmDiagnostic *Summary Method for* exdqlmDiagnostic *Objects*

Description

Summary Method for exdqlmDiagnostic Objects

Usage

```
## S3 method for class 'exdqlmDiagnostic'
summary(object, ...)
```

Arguments

object	An exdqlmDiagnostic object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,mean(y),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.95),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
M0.diags = exdqlmDiagnostics(M0,plot=FALSE)
summary(M0.diags)
```

---

summary.exdqlmForecast

Summary Method for exdqlmForecast *Objects*

Description

Summary Method for exdqlmForecast Objects

Usage

```
## S3 method for class 'exdqlmForecast'
summary(object, ...)
```

Arguments

object	An exdqlmForecast object.
...	Additional arguments (unused).

Examples

```
y <- scIVTmag[1:100]
model <- polytrendMod(1, stats::quantile(y, 0.85), 10)
M0 <- exdqlmISVB(y, p0 = 0.85, model, df = c(0.98), dim.df = c(1),
                  gam.init = -3.5, sig.init = 15)
M0.forecast = exdqlmForecast(start.t = 90, k = 10, m1 = M0)
summary(M0.forecast)
```

summary.exdqlmISVB *Summary Method for* exdqlmISVB *Objects*

Description

Summary Method for exdqlmISVB Objects

Usage

```
## S3 method for class 'exdqlmISVB'
summary(object, ...)
```

Arguments

object	An exdqlmISVB object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M0 = exdqlmISVB(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                  gam.init=-3.5,sig.init=15)
summary(M0)
```

summary.exdqlmMCMC *Summary Method for* exdqlmMCMC *Objects*

Description

Summary Method for exdqlmMCMC Objects

Usage

```
## S3 method for class 'exdqlmMCMC'
summary(object, ...)
```

Arguments

object	An exdqlmMCMC object.
...	Additional arguments (unused).

Examples

```
y = scIVTmag[1:100]
model = polytrendMod(1,quantile(y,0.85),10)
M2 = exdqlmMCMC(y,p0=0.85,model,df=c(0.98),dim.df = c(1),
                gam.init=-3.5,sig.init=15,
                n.burn=100,n.mcmc=150)
summary(M2)
```

`transfn_exdqlmISVB` *Transfer Function exDQLM - ISVB algorithm*

Description

The function applies an Importance Sampling Variational Bayes (ISVB) algorithm to estimate the posterior of an exDQLM with exponential decay transfer function component.

Usage

```
transfn_exdqlmISVB(
  y,
  p0,
  model,
  X,
  df,
  dim.df,
  lam,
  tf.df,
  fix.gamma = FALSE,
  gam.init = NA,
  fix.sigma = TRUE,
  sig.init = NA,
  dqlm.ind = FALSE,
  exps0,
  tol = 0.1,
  n.IS = 500,
  n.samp = 200,
  PriorSigma = NULL,
  PriorGamma = NULL,
  tf.m0 = rep(0, 2),
  tf.C0 = diag(1, 2),
  verbose = TRUE
)
```

Arguments

y	A univariate time-series.
p0	The quantile of interest, a value between 0 and 1.
model	List of the state-space model including GG, FF, prior parameters m0 and C0.
X	A univariate time-series which will be the input of the transfer function component.
df	Discount factors for each block.
dim.df	Dimension of each block of discount factors.
lam	Transfer function rate parameter lambda, a value between 0 and 1.
tf.df	Discount factor(s) used for the transfer function component.
fix.gamma	Logical value indicating whether to fix gamma at gam.init. Default is FALSE.
gam.init	Initial value for gamma (skewness parameter), or value at which gamma will be fixed if fix.gamma=TRUE.
fix.sigma	Logical value indicating whether to fix sigma at sig.init. Default is TRUE.
sig.init	Initial value for sigma (scale parameter), or value at which sigma will be fixed if fix.sigma=TRUE.
dqlm.ind	Logical value indicating whether to fix gamma at 0, reducing the exDQLM to the special case of the DQLM. Default is FALSE.
exps0	Initial value for dynamic quantile. If exps0 is not specified, it is set to the DLM estimate of the p0 quantile.
tol	Tolerance for convergence of dynamic quantile estimates. Default is tol=0.1.
n.IS	Number of particles for the importance sampling of joint variational distribution of sigma and gamma. Default is n.IS=500.
n.samp	Number of samples to draw from the approximated posterior distribution. Default is n.samp=200.
PriorSigma	List of parameters for inverse gamma prior on sigma; shape a_sig and scale b_sig. Default is an inverse gamma with mean 1 (or sig.init if provided) and variance 10.
PriorGamma	List of parameters for truncated student-t prior on gamma; center m_gam, scale s_gam and degrees of freedom df_gam. Default is a standard student-t with 1 degree of freedom, truncated to the support of gamma.
tf.m0	Prior mean of the transfer function component.
tf.C0	Prior covariance of the transfer function component.
verbose	Logical value indicating whether progress should be displayed.

Value

A object of class "exdqlmISVB" containing the following:

- run.time - Algorithm run time in seconds.
- iter - Number of iterations until convergence was reached.

- dqlm.ind - Logical value indicating whether gamma was fixed at 0, reducing the exDQLM to the special case of the DQLM.
- model - List of the augmented state-space model including GG, FF, prior parameters m0 and C0.
- p0 - The quantile which was estimated.
- df - Discount factors used for each block, including transfer function component.
- dim.df - Dimension used for each block of discount factors, including transfer function component.
- lam - Transfer function rate parameter lambda.
- sig.init - Initial value for sigma, or value at which sigma was fixed if fix.sigma=TRUE.
- seq.sigma - Sequence of sigma estimated by the algorithm until convergence.
- samp.theta - Posterior sample of the state vector variational distribution.
- samp.post.pred - Sample of the posterior predictive distributions.
- map.standard.forecast.errors - MAP standardized one-step-ahead forecast errors.
- samp.sigma - Posterior sample of scale parameter sigma variational distribution.
- samp.vts - Posterior sample of latent parameters, v_t, variational distributions.
- theta.out - List containing the variational distribution of the state vector including filtered distribution parameters (fm and fC) and smoothed distribution parameters (sm and sC).
- vts.out - List containing the variational distributions of latent parameters v_t.
- median.kt - Median number of time steps until the effect of X_t is less than or equal to 1e-3.

If dqlm.ind=FALSE, the object also contains:

- gam.init - Initial value for gamma, or value at which gamma was fixed if fix.gamma=TRUE.
- seq.gamma - Sequence of gamma estimated by the algorithm until convergence.
- samp.gamma - Posterior sample of skewness parameter gamma variational distribution.
- samp.sts - Posterior sample of latent parameters, s_t, variational distributions.
- gammasig.out - List containing the IS estimate of the variational distribution of sigma and gamma.
- sts.out - List containing the variational distributions of latent parameters s_t.

Or if dqlm.ind=TRUE, the object also contains:

- sig.out - List containing the IS estimate of the variational distribution of sigma.

Examples

```
y = scIVTmag[1:1095]
X = ELIanoms[1:1095]
trend.comp = polytrendMod(1,mean(y),10)
seas.comp = seasMod(365,c(1,2,4),C0=10*diag(6))
model = trend.comp + seas.comp
M1 = transfn_exdqlmISVB(y,p0=0.85,model=model,
                        X,df=c(1,1),dim.df = c(1,6),
                        gam.init=-3.5,sig.init=15,
                        lam=0.38,tf.df=c(0.97,0.97))
```

Index